Model Predictive Control of Wind Turbines

Henriksen, Lars Christian

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Model Predictive Control of Wind Turbines

Lars Christian Henriksen

Kongens Lyngby 2010
IMM-PHD-2010-244

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Summary

Wind turbines play a major role in the transformation from a fossil fuel based energy production to a more sustainable production of energy. Total-cost-of-ownership is an important parameter when investors decide in which energy technology they should place their capital. Modern wind turbines are controlled by pitching the blades and by controlling the electro-magnetic torque of the generator, thus slowing the rotation of the blades. Improved control of wind turbines, leading to reduced fatigue loads, can be exploited by using less materials in the construction of the wind turbine or by reducing the need for maintenance of the wind turbine. Either way, better total-cost-of-ownership for wind turbine operators can be achieved by improved control of the wind turbines. Wind turbine control can be improved in two ways, by improving the model on which the controller bases its design or by improving the actual control algorithm. Both possibilities have been investigated in this thesis.

The level of modeling detail has been expanded as dynamic inflow has been incorporated into the control design model where state-of-the-art controllers usually assume quasi-steady aerodynamics. Floating wind turbines have been suggested as an alternative to ground-fixed wind turbines as they can be placed at water depths usually thought outside the realm of wind turbine placement. The special challenges posed by controlling a floating wind turbine have been addressed in this thesis.

Model predictive control (MPC) has been the foundation on which the control algorithms have been build. Three controllers are presented in the thesis. The first is based on four different linear model predictive controllers where appropriate switching conditions determine which controller is active. Constraint

handling of actuator states such as pitch angle, pitch rate and pitch acceleration is the primary focus of this controller. The wind turbine is a highly nonlinear plant and a gain scheduling or relinearizing model predictive controller forms the next step to improve performance compared to a linear controller. Finally, a nonlinear model predictive controller has been devised and tested under simplified conditions. At present, the nonlinear model predictive controller is however not expected to be an realistic option for real world application as the computation burden is to heavy to achieve real-time performance.

This thesis is comprised of a collection scientific papers dealing with the various topics presented in this summary.

Resumé

Vindmøller spiller en stor rolle i skiftet fra en fossil-brændstof-baseret energiproduktion til en mere bæredygtig produktion af energi. *Total-cost-of-ownership* er en vigtig parameter, når investorer beslutter i hvilken energi-teknologi de skal placere deres kapital. Moderne vindmøller styres bl.a. ved at *pitche* vingerne og styre generatorens elektro-magnetiske modstand, som bremser vingernes rotation. Forbedret styring af vindmøller, som reducerer udmattelseslaster, kan udnyttes ved at bruge færre materialer i konstruktionen af vindmøller eller ved at reducere behovet for vedligehold af vindmøllerne. Uanset hvad, forbedret *Total-cost-of-ownership* for vindmølleejere kan opnås ved forbedret styring af vindmøllerne. Styring af vindmøller kan forbedres på to måder: Ved at forbedre den model som styringsalgoritmen baseres på eller ved at forbedre den faktiske styringsalgoritme. Begge muligheder er blevet undersøgt i denne afhandling.

Detaljegraden af modellen er blevet udvidet ved at inkludere *dynamisk kølvand* i modellen brugt af styringsalgoritmen, hvor *state-of-the-art* styringer som regel antager "*quasi-stationært kølvands*"-aerodynamik. Flydende vindmøller er blevet foreslået som et alternativ til bund-placerede vindmøller, da de kan placeres på vanddybder normalt antaget udenfor vindmøllers placerings-rækkevidde. De særlige udfordringer som styring af flydende vindmøller giver er blevet undersøgt i denne afhandling.

Model Predictive Control (MPC) har været det fundament som styringsalgoritmerne er blevet bygget på. Tre styringer præsenteres i denne afhandling: Den første er baseret på fire forskellige lineære MPC-styringer, hvor passende skifte-kriterier afgør hvilken styring er aktiv. Håndtering af begrænsinger på aktuatorne, såsom pitch-vinkel, pitch-vinkelhastighed og pitch-vinkelacceleration er det primære fokus for denne styring. Vindmøllen er et stærkt ulineært sys-

tem og *gain-scheduling* eller re-lineærisering er det næste skridt mod forbedret styringsydelse sammenlignet med den lineære styring. Sidst, er en ulineær MPC-styring blevet udviklet og testet under forenklede forhold. For nuværende forventes den ulineære MPC-styring dog ikke at være en realistisk mulighed da den er for beregningstung til at opnå tids-tro ydelse.

Denne afhandling består af en samling af videnskabelige artikler, omhandlende de forskellige emner diskuteret i dette resumé.

Preface and acknowledgments

This Ph.D. thesis has been prepared at DTU Informatics and Risø DTU at the Technical University of Denmark during the period from 2007 to 2010 in partial fulfillment of the requirements of the requirements the Ph.D. degree in engineering. The work has been supervised by Associate Professor Niels Kjølstad Poulsen, DTU Informatics and co-supervised by Senior Scientist Morten Hartvig Hansen, Risø DTU. The work has been funded by Risø DTU.

I am grateful to both of my supervisors for invaluable advise and discussions regarding both control theory and wind turbine modeling. Thanks should also be given to my colleagues from Risø DTU, for their helpful advise regarding wind turbines.

I also wish to thank Associate Professor Carlo Luigi Bottasso, Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, Italy for his hospitality and for all the nice and warm people I met during my stay in Italy. Although no finalized scientific work was produced during the visit in Italy, I gained a lot of inspiration and knowledge which has proved useful in my later work.

Finally, I wish to thank my loving wife Thilde who has been supporting me and who gave birth to our wonderful daughter during the Ph.D studies.

Lyngby, October 2010

Lars Christian Henriksen

Papers included in the thesis

[A] Lars Christian Henriksen, Morten Hartvig Hansen, Niels Kjølstad Poulsen The effect of dynamic inflow in free mean wind speed estimation. *Wind Energy*, 2010. Submitted.

[B] Lars Christian Henriksen, Morten Hartvig Hansen, Niels Kjølstad Poulsen Wind speed and wave force estimation for a floating wind turbine. *Wind Energy*, 2010. Submitted.

[C] Lars Christian Henriksen, Morten Hartvig Hansen, Niels Kjølstad Poulsen Wind Turbine Control with Constraint Handling. *Wind Energy*, 2010. Submitted.

[D] Lars Christian Henriksen, Morten Hartvig Hansen, Niels Kjølstad Poulsen Relinearized Model Predictive Control of a Floating Wind Turbine. *Wind Energy*, 2010. Submitted.

[E] Lars Christian Henriksen, Niels Kjølstad Poulsen, Morten Hartvig Hansen Nonlinear Model Predictive Control of a Simplified Wind Turbine. *Proc. World Congr. Int. Fed. Autom. Control, IFAC*, 2011. Accepted.

[F] Lars Christian Henriksen, Niels Kjølstad Poulsen An online re-linearization scheme suited for Model Predictive and Linear Quadratic Control. IMM-Technical Report 2010-13, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, 2010. Published.

[G] Lars Christian Henriksen, Niels Kjølstad Poulsen A Trust-region-based Sequential Quadratic Programming Algorithm. IMM-Technical Report 2010-14, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, 2010. Published.

Contents

Chapter 1

Introduction

The work presented in the thesis is threefold: wind turbine modeling details (part 1), practical implementation of model predictive control (part 3) and the application of model predictive control on wind turbines (part 2).

The thesis is a collection of papers written during the course of the PhD study. The papers are attached as appendices and form the main matter of this thesis.

This chapter begins with an introduction to wind turbines and wind turbine control. Then follows a list of contributions, which this thesis brings to the field of wind turbine control. The chapter is ended with an outline of the thesis.

1.1 Introduction to wind turbines

Wind turbines exist in many different configurations and sizes: upwind or downwind and horizontal or vertical axis rotors. Upwind means that rotor is placed in front of the tower and downwind means that rotor is placed behind the tower relative to the wind direction. Horizontal or vertical axis refers to around which axis the blades rotate.

A historical view [1, 2] on the field starts with wind mills dating several thousand

years back in time. Wind mills have been used to harness the power of the wind and perform some kind of mechanical work e.g. grinding grain to flour, pumping water for irrigation of fields or pumping water in wet areas to claim land. All these applications converted the mechanical energy from the wind to mechanical energy useful to people. Electrification changed the world and also wind mills. In stead of converting the mechanical energy of the wind to another kind of mechanical energy, the energy was with the use of electrical generators converted to electric energy and wind turbines came into being. The difference between wind mills and wind turbines is that the former produces mechanical energy and the latter produces electrical energy. The electrification of wind mills a century ago was pioneered by the danish scientist, inventor and educationalist Poul la Cour, who also studied the storage of energy using hydrogen, once again a hot topic. Many different configurations with either vertical or horizontal axis rotors have been tested through the years but the predominant configuration today, known as the *danish-type*, is a horizontal axis rotor upwind configuration. Fig. 1.1 depicts a *danish-type* wind turbine. Throughout this thesis it is implied that wind turbine indicates *danish-type* wind turbine.

The fundamentals of the wind turbine can be described by only a few equations. The power captured by the rotor of the wind turbine and transmitted to the drive-shaft is given by

$$P = \frac{1}{2}\rho\pi R^2 V^3 C_P(\theta, V, \Omega) \tag{1.1}$$

where ρ is air mass density, R is rotor radius, V is wind speed, θ is blade pitch angle, Ω is rotor speed and $C_P(\cdot)$ is the power coefficient, describing how much of the power available in the wind is captured by the rotor. The wind turbine rotor speed depends on the aerodynamic torque $Q = P/\Omega$ in the rotor-end of the drive-shaft and on the electro-magnetic generator torque Q_g in the generator-end of the drive-shaft

$$\dot{\Omega} = \frac{1}{J_t}(Q - N_g Q_g) \tag{1.2}$$

where J_t is total inertial mass of the rotor, drive-shaft and generator. N_g is the gear ratio, if a gear is present in the drive-train configuration. The electrical power of the generator is given by

$$P_e = Q_g N_g \Omega \tag{1.3}$$

This simple model can be extended by adding further degrees of freedom. The drive-shaft, initially assumed rigid, is flexible and a torsional degree of freedom could be added to the model to account for this. The tower on which the nacelle is placed is also flexible and fore-aft and side-side displacement of the nacelle due to tower flexibility could also be added to the model. This flexibility means

Figure 1.1: *Danish-type* upwind horizontal axis rotor wind turbine. The three blades are known as the rotor, which is connected to the generator through a drive-shaft, all housed in the nacelle. The nacelle is mounted on top of the tower with a yaw drive enabling the wind turbine to turn towards the wind when the wind changes direction. (Photo by Dirk Ingo Franke and reprint permitted under the CC-BY-SA license.)

that the rotor moves and a relative wind speed, different from the free wind speed, is felt by the rotor. The blades are also flexible and edgewise, flapwise and torsional degrees of freedom could also be included in the model to a more realistic description of the wind turbine. Further degrees of freedom both in structural and aerodynamic respect can be added to further heighten the quality of the model. Dynamic inflow and dynamic stall, which describe how the aerodynamic forces affect blades are also relevant degrees of freedom to add to the model.

Many simulation environments being able to simulate the behavior of wind turbines exist. FAST [3] by the National Renewable Energy Laboratory (NREL) is freely available and is able to produce linearized models of the wind turbine, which is useful for the synthesis of model-based controllers. For a comparison of more aero-elastic codes Passon et al. [4] can be consulted. Throughout this thesis the aero-elastic code HAWC2 [5] developed by Risø DTU has been used to evaluate the performance of the developed controllers.

1.2 Introduction to wind turbine control

Control of wind turbines is not a new field. With the commercialization of pitch controlled variable speed wind turbines in the 1990s, which gradually replaced the predominant stall controlled fixed speed wind turbines, new control challenges emerged. Academic work regarding the control of pitch controlled variable speed wind turbines is discussed in the following section. The works cited in the following are not the only works in existence but have been cited as they represent some of the different methods to control a wind turbine. For a more comprehensive overview of control methods applied to wind turbines Pao and Johnson [6] and Laks et al. [7] can be consulted.

1.2.1 Classic control methods

The first approach to control a wind turbine is to consider it as a single-input-single-output (SISO) system. Leithead and Connor [8] discuss different strategies concerning the control of wind turbines in different operating regimes, where emphasis is more on general concerns rather than the actual synthesis of a particular control method. For above rated wind speeds the pitch of the blades regulate the rotational speed of the wind turbine, as the wind turbine dynamics are nonlinear and change for different wind speeds a gain scheduling scheme can used to achieve similar closed-loop performance for a range of wind speed, such

a control method is described in e.g. Hansen et al. [9]. Dynamic filters might be inserted in the closed-loop to avoid excitation of certain structural modes, such as drive train flexibility or edgewise vibration of the blades. Typically the generator power is controlled by the electro magnetic generator torque in a separate loop. The pitching of blades introduce changes in the thrust force affecting the wind turbine, leading to displacement of the tower top, which in turn leads to changes in relative wind speed felt by the rotor, wind in turn causes the to controller to change the blade pitch angle. Unless the closed-loop poles of the system are placed below the first tower bend frequency unstable behavior can be observed. Bossanyi [10] describes how this can be remedied by adding a new loop to the controller where the tower top acceleration is fed back to the controller and superimposed on the pitch control signal. Cyclic and individual pitch can be used to alleviate the periodic loading caused by the wind shear, tower shadow, turbulent wind field etc. Larsen et al. [11] describes how to add yet another control loop to the controller feeding back pitot tube measurements from the blades to enable individual pitch control. Ever more control loops can be added to dampen the side-side motion of the tower etc. There are no guarantees that the increasingly complicated web of loops in the controller do not interact with each other in undeseriable ways, unless the different loops are separable, indicating there is a potential in the implementation of modern multiple-input-multiple-output (MIMO) control methods.

1.2.2 Modern control methods

The wind turbine is in effect a MIMO system and state space model-based control algorithms are suited to handle such systems. Wright and Balas [12] discuss how to obtain a linearized state space model of the system at a given operating point, using the aero-elastic software FAST [3]. The linearized model can be used to design a linear quadratic (LQ) controller, which is able to handle conflicting control objectives by prioritizing them in a cost function. Selvam et al. [13] uses a multi-blade coordinate transformation to enable individual pitch control within the LQ framework. Stol and Balas [14] also use a LQ framework where periodic disturbance accommodation is used to achieve individual pitch control of the wind turbine. The operation at different wind speeds require some kind of gain scheduling. A theoretically well founded method known as linear parameter varying control (LPV) is seen in the works of Bianchi et al. [15] and Østergaard et al. [16] and provide robust controllers suited for a wide wind speed range. Different nonlinear approaches to the control of wind turbines have also been investigated, among those are Kumar and Stol [17] and Boukhezzar et al. [18]

The use of model predictive control (MPC) is often motivated by its inherent

ability to handle constraints, but nonlinear model predictive control (NMPC) is also useful because it is able to cope with the nonlinear dynamics of the wind turbine. Trainelli et al. [19] explorers the performance of a NMPC algorithm at the presence of an extreme operating gust. Another feature of the MPC and NMPC is the ability to include information about the future into the prediction horizon. Santos [20] explores the potential benefit of including the knowledge of future wind speeds, obtained with e.g. a LIDAR, into the NMPC algorithm.

1.3 Contributions

The contributions and brief descriptions are listed in the following:

- A simplified model of dynamic inflow useful for implementation in model-based control and state estimation algorithms. Presented in paper A and used in papers B and D.
- Wave force inclusion in control design model for a floating wind turbine is in paper B shown to be important for wind speed estimations. Also used in paper D.
- Constraint handling of actuators using model predictive control is presented in paper C.
- Switching conditions for different wind speed regions, ensuring full wind speed range control, is discussed in paper C.
- Relinearization of the model predictive controller to handle nonlinearities of a wind turbine is shown in paper D and derivations of the underlying equations are given in Paper F.
- Nonlinear model predictive control and the benefit of having knowledge of future wind speeds using e.g. LIDARs is investigated in Paper E. Papers F and G describe the underlying equations and methods.

1.4 Outline of thesis

Chapter 2, 3 and 4 summarizes the papers concerning wind turbine modeling, the papers concerning implementation of model predictive control algorithms and the papers concerning implemented wind turbine controllers, respectively.

Chapter 5 draws conclusion regarding the work presented in this thesis.

Appendices A and B comprises the first part. The papers deal with different aspects of wind turbine modeling used by model-based control algorithms.

Appendices C, D and E comprises the second part. These papers describe different implementations of model predictive control applied on wind turbines.

Appendices F and G comprises the third part. The papers in this part describes in detail the underlying methods of the model predictive control algorithms presented in the second part.

Chapter 2

Wind Turbine Modeling

Model-based control and state estimation algorithms require control design model of sufficient quality both in the inclusion of all significant degrees of freedom and in the calculation/estimation of model parameters. With this in mind, the Papers A and B examine what importance various degrees of freedom have on the wind speed estimation. The investigations of wind speed estimation performance are motivated by the naïve assumption that if wind speed estimation is improved, then control performance also has a potential to be improved.

2.1 Paper A: The effect of dynamic inflow in free mean wind speed estimation

An often neglected degree of freedom in terms of control design modeling is dynamic inflow. The term dynamic inflow describes the phenomena that the aerodynamic forces do not settle instantly to a new equilibrium when changes in wind speed, rotor speed or blade pitch angle occur. Fig. 2.1 shows a cross section of a blade, at radial distance r from the center of the rotor, and shows the local aerodynamic forces at that point. The local induced velocities $v_n(r) = Va_n$ and $v_t(r) = \Omega r$, in normal and tangential directions respectively, will under constant conditions settle to an equilibrium over a period of time. The equilibrium points

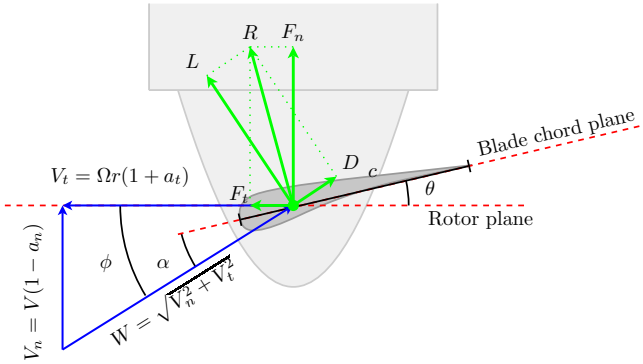


Figure 2.1: Cross section of blade in the span wise direction along the blade (Paper A).

for $v_n(r)$ and $v_t(r)$ can be determined offline, giving a C_P curve used by (1.1) which gives the quasi-steady aerodynamic torque

$$Q(V, \Omega, \theta) = \frac{\frac{1}{2}\rho\pi R^2 V^3 C_P(V, \Omega, \theta)}{\Omega} \tag{2.1}$$

A better description of the aerodynamic forces is to determine the instant induced velocities online, where n_B is the number of blades on the rotor. The instant aerodynamic torque is given by

$$Q(V, \Omega, \theta, v_n(r), v_t(r)) = n_B \int_0^R r F_t(V, \Omega, \theta, v_n(r), v_t(r), r) dr \tag{2.2}$$

where the induced velocities are calculated using one of the different models describing dynamic inflow in the work of Snel and Schepers [21].

Although van Engelen and van der Hooft [22] have demonstrated dynamic inflow to be of importance, quasi-steady aerodynamics are usually used in the control design models. Paper A presents a simplified dynamic inflow model based on a more detailed model by Øye [23]. The presented model assumes quasi-steady tangential induced velocities and determines the induced axial velocities based on the quasi-steady distribution of axial induction factors a_n^{qs} divided by the quasi-steady distribution of axial induction factors $\bar{a}_n^{qs}$ averaged along the blade span. The normalized quasi-steady distribution is then multiplied with an axial induced wind speed $\bar{v}_n$ averaged along the blade span, giving

$$v_n(r) = \frac{a_n^{qs}(V, \Omega, \theta, r)}{\bar{a}_n^{qs}(V, \Omega, \theta)} \bar{v}_n \tag{2.3}$$

The only state which should be added to the control design model is $\bar{v}_n$ with temporal dynamics given by

$$\bar{v}_n = \frac{1}{\tau s + 1} V \bar{a}_n^{qs}(V, \Omega, \theta) \tag{2.4}$$

with the time constant

$$\tau = \frac{1}{2} \frac{1.1R}{V - 1.3\bar{v}_n} \tag{2.5}$$

The presented model is suitable for inclusion in control design models and Fig. 2.2 shows how wind speed estimation is improved by the inclusion of the simplified dynamic inflow model compared to assuming quasi-steady aerodynamics.

2.2 Paper B: Wind speed and wave force estimation for a floating wind turbine

Floating wind turbines in various configurations are discussed by e.g. Jonkman [24]. The spar buoy concept, also examined by Skaare et al. [25], is investigated in Paper B. The floating wind turbine concept is highly sensitive to the aerodynamic thrust

$$T(V_{rel}, \Omega, \theta, \bar{v}_n) = n_B \int_0^R F_n(V_{rel}, \Omega, \theta, v_n(r), r) dr \tag{2.6}$$

where the relative wind speed felt by the rotor is given by the difference between the free wind speed and the fore-aft velocity of the tower top

$$V_{rel} = V - \dot{y}_{top} \tag{2.7}$$

Figure 2.3 shows that the pitch angle of the top of the wind turbine and the pitch angle of the bottom of the wind turbine are in phase, but a closer look best seen in the acceleration signal $\ddot{y}$, shows that top and bottom bottom angles also have an out of phase contribution due to the tower bend degree of freedom.

The structural model of the floating wind turbine in the fore-aft direction is given by the interconnected two mass-damper-spring systems (M_{tp}, D_{tp}, K_{tp}) and (M_{tb}, D_{tb}, K_{tb}), respectively describing tower pitch and tower bend degrees of freedom.

$$M_{tb}\ddot{y}_{rel} + D_{tb}\dot{y}_{tb} + K_{tb}y_{tb} = T \tag{2.8a}$$

$$\tilde{M}_{tp}\ddot{y}_{tp} + D_{tp}\dot{y}_{tp} + K_{tp}y_{tp} = F_w + D_{tb}\dot{y}_{tb} + K_{tb}y_{tb} \tag{2.8b}$$

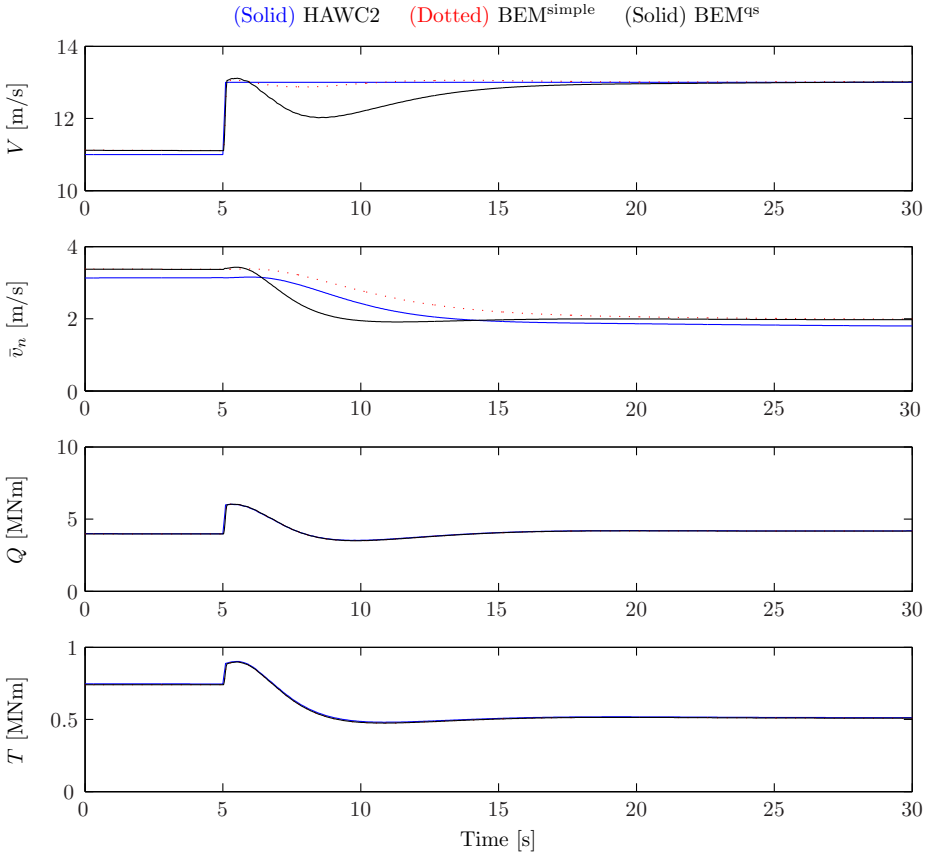


Figure 2.2: Estimation of free mean wind speed and averaged axial induced velocity and aerodynamic forces. The extended Kalman filter based on the BEM^{qs} model fails to predict the wind speed transient seen from time 5 s to 20 s, whereas the extended Kalman filter based on the $\text{BEM}^{\text{simple}}$ model, which takes the dynamic inflow into account, captures the behavior (Paper A).

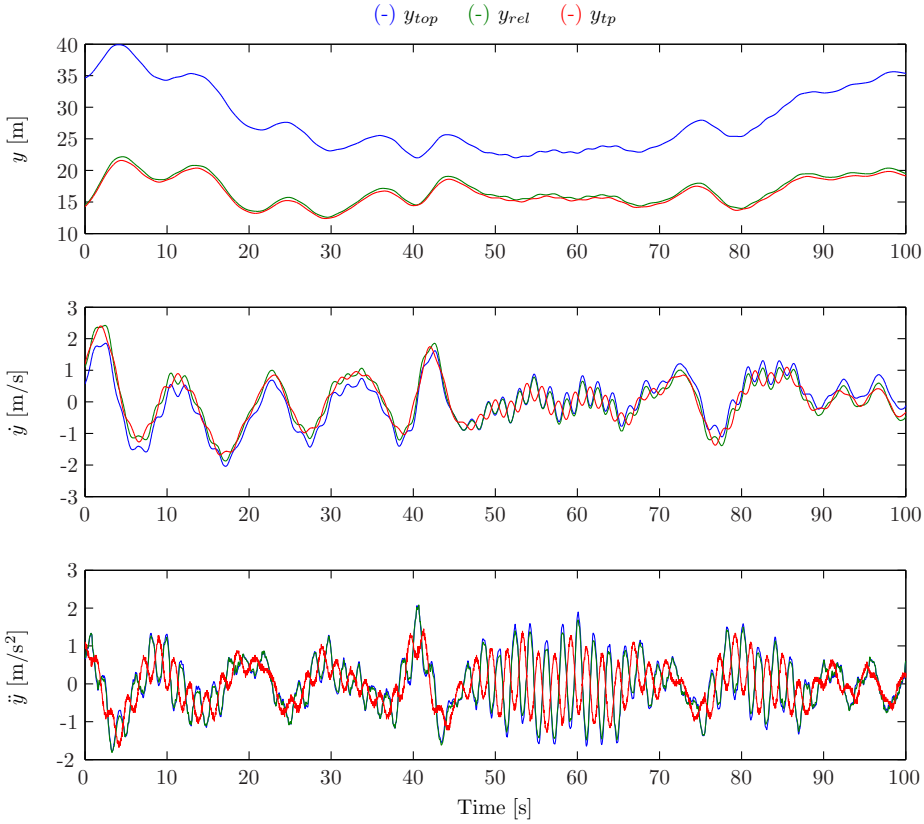


Figure 2.3: Floating tower fore-aft motion - The first plot shows tower top position shown with 3 different sensors: 1. absolute position in y direction, 2. position relative to position of bottom of floater in y direction, 3. absolute pitch angle multiplied with height of wind turbine. The second plot depicts the velocities of the 3 sensors. It can be seen that low frequency displacement of the entire structure (surge) has little impact on the velocity measurements. The third plot display the accelerations and it can be seen that the tower bending mode causes the top and bottom sensors to differ (Paper B).

where turbine pitch displacement is $y_{tp} = h \sin \alpha \approx h\alpha$. Tower bend displacement is given by $y_{tb} = y_{rel} - y_{tp}$.

The term F_w in (2.8b) approximates the influence of wave forces on the on the floating structure. Paper B shows that the influence of wave forces should be included in the control design model to avoid that nacelle displacements are solely ascribed to the aerodynamic thrust when in fact wave forces also play a major role. If not dealt with, erroneous wind speed estimates are produced by the state estimation algorithm. In Fig. 2.4 it can be seen that wave forces (proportional to the water acceleration $\dot{u}$) as well as tower bending are important degrees of freedom that should be included in the control design model to achieve good wind speed estimates. Furthermore, it can be speculated that wave forces should also be included in control design models for offshore wind turbines with ground-fixed foundations.

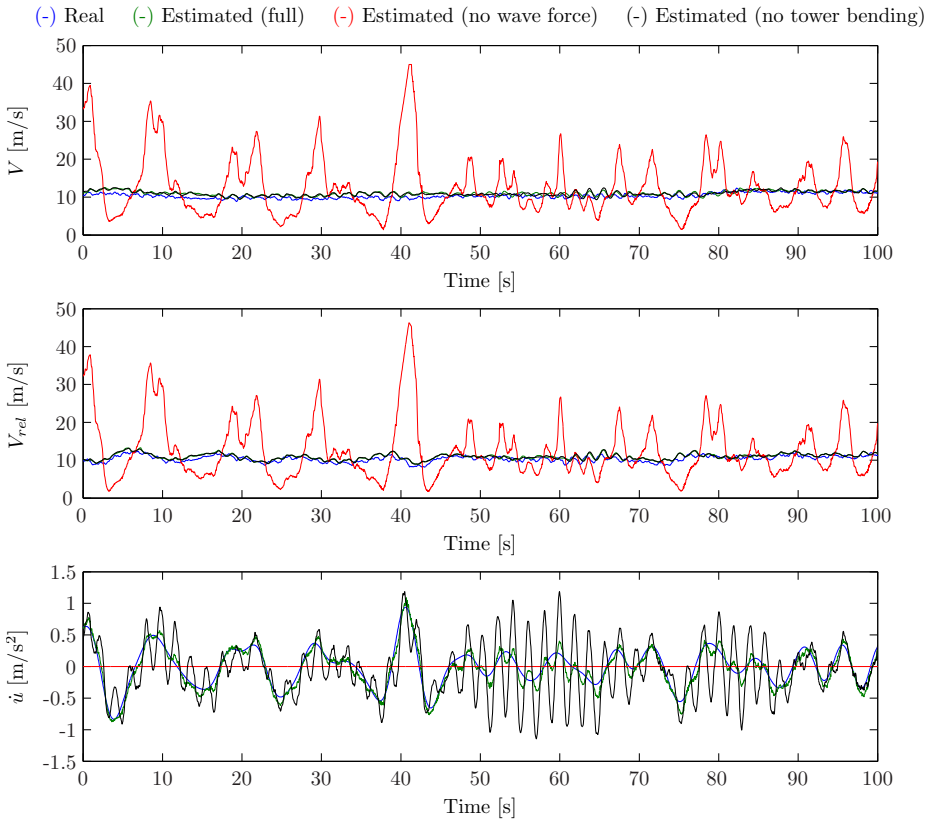


Figure 2.4: Wind speed and wave force estimation by extended Kalman filters based on three different models: The first model includes all the submodels presented in this work, including tower bending and wave force contribution. The second model omits wave forces and the third model excludes tower bending (Paper B).

Chapter 3

Numerical methods

Papers C, D and E all use Model Predictive Control in different implementations. Where Paper C represents a typical linear MPC implementation and Paper E uses a nonlinear MPC, then Paper D is an intermediate step between Papers C and E, as the controller in Paper D is relinearized in each sample, but assumes the system to be linear within the prediction horizon.

A basic formulations of MPC, where e.g. soft constraints are not included, with a time-discrete prediction horizon of length N is

$$\min \phi_N(\boldsymbol{x}_N, \boldsymbol{u}_N) + \sum_{k=0}^{N-1} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{3.1a}$$

subject to

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}} \tag{3.1b}$$

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{x}_{k+1} = \boldsymbol{0}, \qquad k = 0, \ldots, N \tag{3.1c}$$

$$\boldsymbol{c}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0}, \qquad k = 0, \ldots, N \tag{3.1d}$$

where (3.1a) contains the stage-wise cost function $\phi_k(\cdot)$ which should be minimized. Terms of the cost function to be minimized, could in the case of a wind turbine be weighted and squared terms containing blade pitch action, nacelle movement and generator speed and power output deviations from nominal values etc.. The optimization problem is subject to the initial condition constraint

(3.1b) where $\bar{\boldsymbol{x}}$ is the current measurement/estimate of the state vector. The nonlinear state progress equation (3.1c) is a based on the control design model, which should be an *as-good-as-possible* approximation of the real plant (wind turbine). The control problem could be subject to constraints (3.1d) such as pitch and generator torque rate limitations. Solving (3.1) yields the predicted optimal sequences of states

$$\vec{\boldsymbol{x}} = [\boldsymbol{x}_0^T \ \boldsymbol{x}_1^T \ \dots \ \boldsymbol{x}_N^T]^T \tag{3.2}$$

and control signals

$$\vec{\boldsymbol{u}} = [\boldsymbol{u}_0^T \ \boldsymbol{u}_1^T \ \dots \ \boldsymbol{u}_{N-1}^T]^T \tag{3.3}$$

where $\boldsymbol{u}_0$ is actuated. The total MPC problem (3.1) is then solved in each sample with updated measurements/estimates of the current state $\boldsymbol{x}_0$.

The state progress equation (3.1c) can be assumed to be a constant linear model within the prediction horizon to reduce the computational burden and to reduce the overall complexity of the controller. Paper C approximates (3.1c) as a linear system linearized around an equilibrium point ($\boldsymbol{x}_* = \underline{\boldsymbol{f}}(\boldsymbol{x}_*, \boldsymbol{u}_*)$) depending on a specific wind speed

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \boldsymbol{x}_* + \mathbf{A}_*(\boldsymbol{x}_k - \boldsymbol{x}_*) + \mathbf{B}_*(\boldsymbol{u}_k - \boldsymbol{u}_*) \tag{3.4}$$

Paper D allows for more flexibility as (3.1c) can be linearized at arbitrarily chosen points $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ and not only equilibrium points

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \underline{\boldsymbol{f}}(\boldsymbol{x}_0, \boldsymbol{u}_0) + \mathbf{A}_0(\boldsymbol{x}_k - \boldsymbol{x}_0) + \mathbf{B}_0(\boldsymbol{u}_k - \boldsymbol{u}_0) \tag{3.5}$$

Finally, Paper E makes no assumption of constant linear behavior within the prediction horizon and uses stage-wise linearizations of the state progress equation in the optimization routine

$$\underline{\boldsymbol{f}}(\Delta\boldsymbol{x}_k + \boldsymbol{x}_k, \Delta\boldsymbol{u}_k + \boldsymbol{u}_k) \approx \underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{A}_k\Delta\boldsymbol{x}_k + \mathbf{B}_k\Delta\boldsymbol{u}_k \tag{3.6}$$

where $(\Delta\boldsymbol{x}_k, \Delta\boldsymbol{u}_k)$ are iterative steps within the nonlinear optimization algorithm. For each iteration i the current variables are updated

$$(\boldsymbol{x}_k^{i+1}, \boldsymbol{u}_k^{i+1}) = (\Delta\boldsymbol{x}_k, \Delta\boldsymbol{u}_k) + (\boldsymbol{x}_k^i, \boldsymbol{u}_k^i) \tag{3.7}$$

until some specified convergence criteria are met.

3.1 Paper F: An online re-linearization scheme suited for Model Predictive and Linear Quadratic Control

The total MPC problem (3.1) forms a nonlinear programming problem (NLP) which is computationally heavy to solve. Solving techniques will typically approximate the NLP as a quadratic programming problem (QP) and the QP will be updated in iterations towards the optimal nonlinear solution. The QP in its full form is given by

$$\begin{aligned}\min \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_N \\ \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix}^T & \begin{bmatrix} \mathbf{Q}_0 & & & \mathbf{M}_0^T & & \\ & \ddots & & & \ddots & \\ & & \mathbf{Q}_N & & & \\ \mathbf{M}_0 & & & \mathbf{R}_0 & & \\ & \ddots & & & \ddots & \\ & & & & & \mathbf{R}_{N-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_N \\ \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix} \\ & + \begin{bmatrix} \boldsymbol{q}_0^T & \dots & \boldsymbol{q}_N^T & \boldsymbol{r}_0^T & \dots & \boldsymbol{r}_{N-1}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_N \\ \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix} \end{aligned} \tag{3.8a}$$

s.t.

$$\begin{bmatrix} \mathbf{I} & & \dots & & & \dots & \\ \mathbf{A}_0 & -\mathbf{I} & \dots & \mathbf{B}_0 & & \dots & \\ & \mathbf{A}_1 & \dots & & \mathbf{B_1} & \dots & \\ & & \vdots & & & \vdots & \\ & & \dots & -\mathbf{I} & & \dots & \mathbf{B}_{N-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \\ \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{x}} \\ -\boldsymbol{b}_0 \\ -\boldsymbol{b}_1 \\ \vdots \\ -\boldsymbol{b}_{N-1} \end{bmatrix} \tag{3.8b}$$

$$\begin{bmatrix} \mathbf{C}_0 & & \dots & \mathbf{D}_0 & & \dots & \\ & \mathbf{C}_1 & \dots & & \mathbf{D}_1 & \dots & \\ & & \vdots & & & \vdots & \\ & & \dots & & & \dots & \mathbf{D}_{N-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \\ \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix} \leq \begin{bmatrix} -\boldsymbol{d}_0 \\ -\boldsymbol{d}_1 \\ \vdots \\ -\boldsymbol{d}_{N-1} \end{bmatrix} \tag{3.8c}$$

where $\mathbf{Q}_k$, $\mathbf{M}_k$ and $\mathbf{R}_k$ are Hessians of the stage-wise cost function and $\boldsymbol{q}_k$ and $\boldsymbol{r}_k$ are gradients of the stage-wise cost function. $\mathbf{A}_k$ and $\mathbf{B}_k$ are Jacobians of the stage progress equation and $\mathbf{C}_k$ and $\mathbf{D}_k$ are Jacobians of the inequality constraint. The constant terms $\boldsymbol{b}_k$ and $\boldsymbol{d}_k$ stems from the linearization of the state progress and inequality constraint equations, respectively.

The matrices of the MPC problem (3.8) are sparse, which should be exploited to reduce the computational cost. Rao et al. [26] demonstrated that the QP given by a MPC problem can be reformulated to a time-discrete Riccati equation rendering the computational burden dependant on the prediction horizon length in linear way $O(N)$ rather than cubic $O(N^3)$. This is of particular high importance when long prediction horizons are used. Paper F documents the specific implementation exploiting the Riccati structure of the MPC problem. The implemented solver presented in Paper F is used by the controllers in Papers C, D and E.

3.2 Paper G: A Trust-region-based Sequential Quadratic Programming Algorithm

Paper G, inspired by the work of Tenny et al. [27], describes how the nonlinear model predictive control problem in Paper E is solved using a trust-region-based sequential quadratic programming problem algorithm. The general NLP [28, 29] is given by

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}), \text{ s.t. } \boldsymbol{c}(\boldsymbol{x}) = \mathbf{0} \text{ and } \boldsymbol{d}(\boldsymbol{x}) \leq= \mathbf{0} \tag{3.9}$$

which can be approximated as a quadratic programming problem

$$\min_{\Delta \boldsymbol{x}} \boldsymbol{m}(\Delta \boldsymbol{x}), \ \boldsymbol{m}(\Delta \boldsymbol{x}) = \tfrac{1}{2} \Delta \boldsymbol{x}^T \mathbf{H} \Delta \boldsymbol{x} + \nabla \boldsymbol{f}(\boldsymbol{x})^T \Delta \boldsymbol{x} \tag{3.10a}$$

subject to

$$\boldsymbol{c}(\boldsymbol{x}) + \nabla \boldsymbol{c}(\boldsymbol{x})^T \Delta \boldsymbol{x} = \mathbf{0} \tag{3.10b}$$

$$\boldsymbol{d}(\boldsymbol{x}) + \nabla \boldsymbol{d}(\boldsymbol{x})^T \Delta \boldsymbol{x} \leq \mathbf{0} \tag{3.10c}$$

$$\|\mathbf{D}\Delta \boldsymbol{x}\|_p \leq \delta \tag{3.10d}$$

where the optimization variables are updated in each iteration, if certain step acceptance criteria are met. The trust-region inequality constraint (3.10d) which is tightened or loosened depending of the progress of iterations. If the approximated QP (3.10) is a good description of the original NLP (3.9) then the solution obtained by solving the QP can be trusted and large steps can be taken in each iteration and the trust-region radius should be large. If the NLP is not well approximated by the QP then the solutions obtained by solving the QP are not to be trusted and small steps should be taken and the trust-region radius should be reduced. The QP and NLP are typically not in good agreement if the NLP is highly nonlinear or if the optimization problem is not convex.

The trust-region scaling matrix $\mathbf{D}$ should somehow represent the optimization problem such that the the dimensions of the trust-region reflect the sensitivity of the cost function with regards to the optimization variable. A decomposition of the Hessian of the cost function $\mathbf{H}$ can be used to form the scaling matrix $\mathbf{D}$ and Paper G can be consulted for further details. Fig. 3.1 depicts examples of ∞-norm and 2-norm trust regions based on the Hessian of the optimization problem.

When applying the trust-region-based sequential quadratic programming problem algorithm to NMPC, the special structure of the NMPC should be exploited to reduce the computational cost. The underlying QP is solved using the specialized QP solver from Paper F and calculations of Hessians, gradients, scaling matrices for the trust-region etc. also exploit the sparse structure of the NMPC optimization problem.

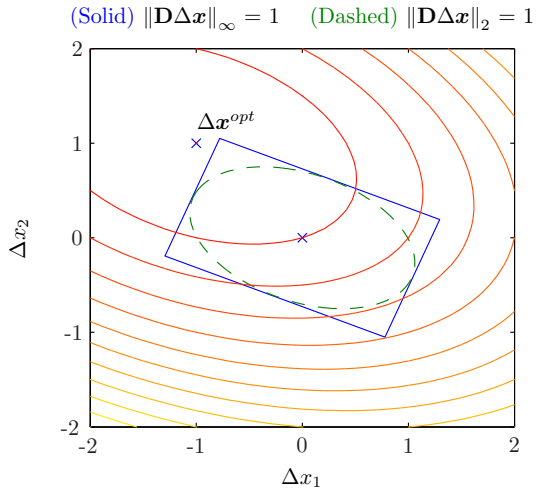


Figure 3.1: Two trust regions are shown in this figure: The box is given by the ∞-norm and the ellipsoid is given by the 2-norm. Both norms have scaling matrices based on the Hessian of the cost function (Paper G).

Chapter 4

Implemented controllers

The simplified description of the wind turbine, given by (1.1) to (1.3), can be used to illustrate the overall control objectives for the wind turbine. The overall primary control objective of the wind turbine for a given wind speed can be formulated as

$$\min(P_e - P_{nom})^2 \tag{4.1a}$$

subject to the steady state equality constraint

$$0 = \frac{1}{J_t}(Q - N_g Q_g) \tag{4.1b}$$

A requirement for the generator speed to be within a specified operating range gives the following constraint

$$\Omega \in (\Omega_{min}, \Omega_{nom}) \tag{4.1c}$$

where P_{nom} is the nominal/rated power of the generator and Ω_{nom} is the nominal/rated speed of the generator. For below rated wind speeds, where $P_e < P_{nom}$, there is a unique solution for each wind speed. For above rated wind speeds, where $P_e = P_{nom}$, multiple solutions exist and the cost function (4.1a) should be extended to

$$\min(P_e - P_{nom})^2 + (\Omega - \Omega_{nom})^2 \tag{4.1d}$$

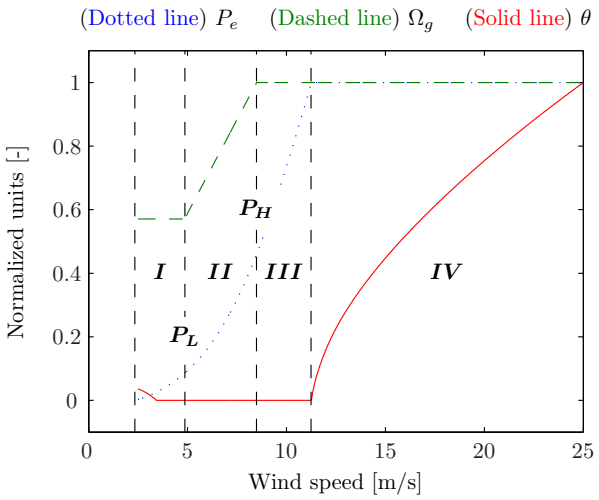


Figure 4.1: Sweep of wind speeds showing the steady state values of the primary variables of the wind turbine. Notice the pitch angle would normally have slightly different values for wind speeds below rated wind speed. The pitch angle in this case have been constrained and is thus different from its unconstrained value. This is however not expected to have significant influence on the performance of the controller (Papers C and D).

to ensure the existence of a unique solution and to ensure that nominal rotational speed is the control target.

Solving (4.1) reveals that wind turbine controllers are hybrid in nature as they operate under different wind speeds with different wind speed specific control objectives. Fig. 4.1 shows how the regions of operation can be divided into regions I to IV where regions I to III are known as partial load regions and region IV is the full load region. Besides of having to achieve good control performance within one of the regions, the controller should also be able to switch between the different regions of operation in a smooth manner. The Papers C, D and E form a development towards having integrated the full wind speed range in a single control law.

Although the overall primary control objective is given by (4.1), Papers C and D use a modified primary control objective in the partial load regions to cope with the nonlinear dynamics of the wind turbine. The collective blade pitch angle is kept at its optimal value and the optimal generator speed for a given below rated wind speed is tracked with the generator torque. The NMPC formulation used in Paper E cancels the need for a reformulation of the primary control objectives

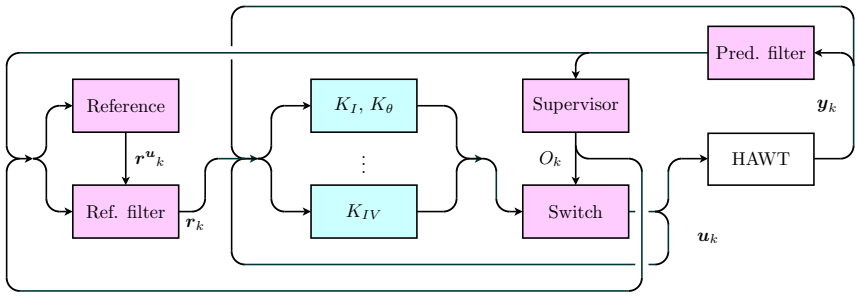


Figure 4.2: Setup of the hybrid controller. Regions I to III are only controlled by the generator torque controllers K_I to K_{III}. The pitch angle is kept constant by a separate controller K_θ in these regions. In region IV both pitch angle and generator torque are used in the same controller (Paper C).

in the partial load regions and (4.1) is used throughout the wind speed range.

4.1 Paper C: Wind Turbine Control with Constraint Handling: A Model Predictive Control Approach

Paper C presents a hybrid controller consisting of four different linear model predictive controllers, one for each region of operation. A set of switching conditions are designed to determine which of the four controllers that should be active. The setup of the hybrid controller used in Paper C can be seen in Fig. 4.2.

The design of the model predictive controllers is heavily inspired by Pannocchia et al. [30] and Pannocchia and Rawlings [31]. The control design model is augmented with a disturbance model $(\mathbf{E}, \mathbf{F})$, with $\boldsymbol{d}$ and $\boldsymbol{p}$ as state and output disturbances, respectively. The disturbance model is used to achieve offset-free performance.

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} \underline{\mathbf{A}} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\hat{\underline{\mathbf{A}}}} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_k + \underbrace{\begin{bmatrix} \underline{\mathbf{B}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\hat{\underline{\mathbf{B}}}} \boldsymbol{u}_k + \begin{bmatrix} \boldsymbol{w}_{\boldsymbol{x}} \\ \boldsymbol{w}_{\boldsymbol{d}} \\ \boldsymbol{w}_{\boldsymbol{p}} \end{bmatrix}_k \tag{4.2a}$$

$$\boldsymbol{y}_k = \underbrace{\begin{bmatrix}\mathbf{C} & \mathbf{0} & \mathbf{F}\end{bmatrix}}_{\hat{\mathbf{C}}} \begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{d}\\ \boldsymbol{p}\end{bmatrix}_k + \mathbf{D}\boldsymbol{u}_k + \boldsymbol{v}_k \tag{4.2b}$$

where

$$\boldsymbol{v} \in N(\mathbf{0}, \mathbf{R_y}) \text{ and } \begin{bmatrix}\boldsymbol{w_x}\\ \boldsymbol{w_d}\\ \boldsymbol{w_p}\end{bmatrix} \in N(\mathbf{0}, \mathbf{R_{xdp}})$$

This is achieved as a combined state and disturbance estimator ascribe the discrepancies between the measured outputs $\boldsymbol{y}$ of the real plant (wind turbine) and the predicted outputs of the of the model-based estimator to the augmented disturbances. Steady state target state and inputs $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$ are calculated via target calculation

$$\begin{bmatrix}\underline{\mathbf{A}} - \mathbf{I} & \underline{\mathbf{B}}\\ \mathbf{HC} & \mathbf{HD}\end{bmatrix} \begin{bmatrix}\bar{\boldsymbol{x}}\\ \bar{\boldsymbol{u}}\end{bmatrix}_k = \begin{bmatrix}-\mathbf{E}\hat{\boldsymbol{d}}\\ \boldsymbol{r} - \mathbf{HF}\hat{\boldsymbol{p}}\end{bmatrix}_k \tag{4.3}$$

where $\boldsymbol{r}$ is the reference to reach in steady state and $\mathbf{H}$ is a matrix that maps the measured outputs to the reference outputs. The calculated target values are then used by the constrained linear quadratic regulator (CLQR) [30, 32] where the origin shifted variables

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}} \text{ and } \tilde{\boldsymbol{u}} = \boldsymbol{u} - \bar{\boldsymbol{u}}$$

are introduced. The CLQR problem (Paper C should be consulted to see a formulation with inclusion of soft constraints and terminal cost) is

$$\min \sum_{k=0}^{N} \frac{1}{2}(\tilde{\boldsymbol{x}}_k^T \mathbf{Q}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{u}}_k^T \mathbf{R}\tilde{\boldsymbol{u}}_k + 2\tilde{\boldsymbol{x}}_k^T \mathbf{M}\tilde{\boldsymbol{u}}_k) \tag{4.4a}$$

subject to

$$\tilde{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}} - \bar{\boldsymbol{x}} \tag{4.4b}$$

$$\tilde{\boldsymbol{x}}_{k+1} = \underline{\mathbf{A}}\tilde{\boldsymbol{x}}_k + \underline{\mathbf{B}}\tilde{\boldsymbol{u}}_k \qquad k = (0, 1, \ldots, N-1) \tag{4.4c}$$

$$\mathbf{C_h}\tilde{\boldsymbol{x}}_k + \mathbf{D_h}\tilde{\boldsymbol{u}}_k \leq \boldsymbol{h} \qquad k = (0, 1, \ldots, N) \tag{4.4d}$$

A sequence of states and control signals are calculated when solving the QP formed by the MPC problem. The first origin shifted control signal $\tilde{\boldsymbol{u}}_0$ in the computed sequence is added to the target control signal $\bar{\boldsymbol{u}}$ and the sum gives the actual control signal $\boldsymbol{u}$. The setup of disturbance and state estimator, target calculation and constrained linear quadratic regulator can be seen in Fig. 4.3.

Paper C explores the constraint handling capabilities of the CLQR during an extreme operating gust where actuator rate limits have been set to more restrictive limits to ensure that limits where reached during the simulation in an effort to challenge the controllers, results for the simulation can be seen in Fig. 4.4.

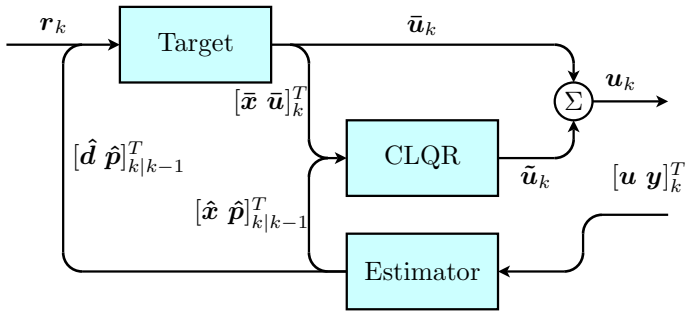


Figure 4.3: Origin shifting model predictive controller setup. Notice that the control signal given by the controller is not directly fed back to the estimator. This is because only the control signal of the active controller in the hybrid controller is fed back to the estimator (Paper C).

4.2 Paper D: Relinearized Model Predictive Control of a Floating Wind Turbine

Paper D represents the next step towards a single control law being able to handle the full wind speed range. The number of regions of operation are reduced from four to two as the partial load regions are combined to a single region of operation aiming to increase power capture. Instead of having four linearized controllers as in Paper C a single controller is relinearized in each sample. Switching conditions then determine if that controller should use partial or full load control objectives. The setup of the hybrid controller used in Paper D can be seen in Fig. 4.5.

The relinearization in each sample enables the linearized control design model used by the controller to be in better agreement with the actual dynamics of the plant at the current operating point. The control design model is augmented with a linear disturbance model $(\mathbf{E}, \mathbf{F})$ as in Paper C, but the control design model is nonlinear

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} \underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{E}\boldsymbol{d}_k \\ \boldsymbol{d}_k \\ \boldsymbol{p}_k \end{bmatrix}}_{\hat{\underline{\boldsymbol{f}}}(\boldsymbol{x}_k, \boldsymbol{d}_k, \boldsymbol{p}_k, \boldsymbol{u}_k)} + \begin{bmatrix} \boldsymbol{w}_{\boldsymbol{x}} \\ \boldsymbol{w}_{\boldsymbol{d}} \\ \boldsymbol{w}_{\boldsymbol{p}} \end{bmatrix}_k \tag{4.5a}$$

$$\boldsymbol{y}_k = \underbrace{\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{F}\boldsymbol{p}_k}_{\hat{\boldsymbol{g}}(\boldsymbol{x}_k, \boldsymbol{d}_k, \boldsymbol{p}_k, \boldsymbol{u}_k)} + \boldsymbol{v}_k \tag{4.5b}$$

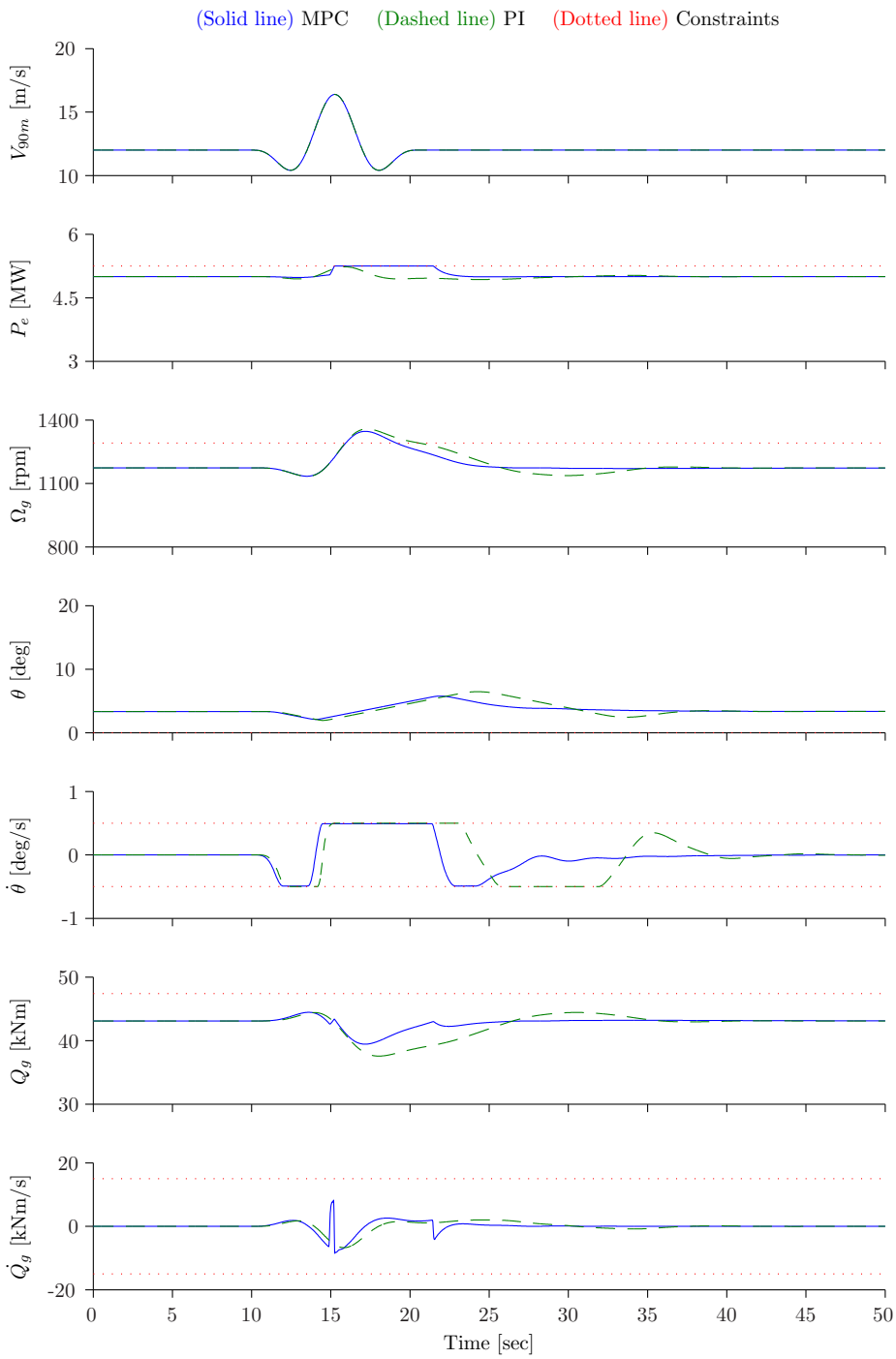


Figure 4.4: Test case 4 with extreme operating gust at a mean wind speed of 12 m/s. Tower shadow and wind shear has been disabled to achieve clean results. The pitch rate constraints has been drastically reduced from 8 deg/s to 0.5 deg/s to challenge the controllers (Paper C).

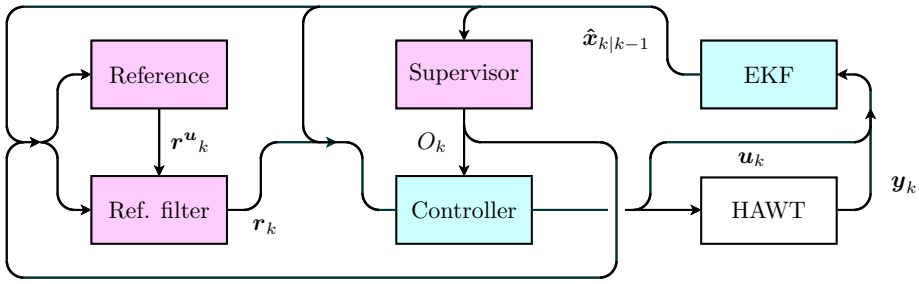


Figure 4.5: Setup of the hybrid controller. An extended Kalman filter provides estimates of states used by other blocks in the diagram. Supervisor block provides partial or full load control objectives to controller depending on switching conditions. Reference and reference filter blocks provide references for the controller to track depending on whether partial or full load operation is active (Paper D).

where

$$\begin{bmatrix} \boldsymbol{w_x} \\ \boldsymbol{w_d} \\ \boldsymbol{w_p} \end{bmatrix} \in N(\mathbf{0}, \mathbf{R_{xdp}}), \text{ and } \boldsymbol{v} \in N(\mathbf{0}, \mathbf{R_y})$$

and states and disturbances are estimated by an extended Kalman filter. The target calculation and origin shifting setup of Paper C is also changed in Paper D to a single formulation where reference tracking is incorporated into an extended LQ/MPC formulation (Paper D should be consulted to see a formulation with inclusion of soft constraints and terminal cost) giving

$$\min \sum_{k=0}^{N} \frac{1}{2} \boldsymbol{g_z}(\boldsymbol{x}_k, \boldsymbol{u}_k)^T \mathbf{W_z} \boldsymbol{g_z}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \frac{1}{2} (\boldsymbol{r} - \boldsymbol{g_r}(\boldsymbol{x}_k, \boldsymbol{u}_k))^T \mathbf{W_r} (\boldsymbol{r} - \boldsymbol{g_r}(\boldsymbol{x}_k, \boldsymbol{u}_k)) \quad (4.6a)$$

subject to

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}} \quad (4.6b)$$

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{x}_{k+1} = \mathbf{0} \qquad k = (0, 1, \ldots, N-1) \quad (4.6c)$$

$$\boldsymbol{g_h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{h} \qquad k = (0, 1, \ldots, N) \quad (4.6d)$$

the nonlinear equations are linearized as described in Papers D and F

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \underline{\mathbf{A}} \boldsymbol{x}_k + \underline{\mathbf{B}} \boldsymbol{u}_k + \boldsymbol{\delta} \quad (4.7)$$

$$\boldsymbol{g_r}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{C_r} \boldsymbol{x}_k + \mathbf{D_r} \boldsymbol{u}_k + \boldsymbol{\gamma_r} \quad (4.8)$$

the output functions $\boldsymbol{g_z}(\cdot)$ and $\boldsymbol{g_h}(\cdot)$ are linearized in similar manner as $\boldsymbol{g_r}(\cdot)$. The estimated disturbances are added to the constant contributions $\boldsymbol{\delta}$ and $\boldsymbol{\gamma_r}$ to achieve offset-free control.

For time-varying systems such as a rotating wind turbine where the individual blade pitch angles do not have a constant steady state value but rather a periodic steady state value, the target calculation of Paper C is not suitable unless multiblade transformations [13] are used. The MPC formulation in Paper D does not suffer from this and individual pitch control could be put into the framework of this controller.

A floating wind turbine is controlled in this paper and the presented controller is compared to a benchmark PI controller [24]. Fatigue load reductions compared to the PI controller are observed in all regions of operation. Results from a simulation with a mean wind speed of 12 m/s can be seen in Fig. 4.6 where the PI controller is compared to two RLMPC controllers: The first assume quasi-steady aerodynamics and the second included dynamic inflow in the control design model. No significant differences between the two control design models can be observed and further investigations are needed.

4.3 Paper E: Nonlinear Model Predictive Control of a Simplified Wind Turbine

Paper E discusses the next step towards a full wind speed range single control law. Nonlinear model predictive control is employed to achieve an even better fit between the control design model and the nonlinear plant dynamics. The formulation is the same as in Paper D, but the nonlinear model is relinearized in each stage of the prediction horizon and not assumed linear within the prediction horizon as in Paper D. Papers F and G documents the underlying algorithms of the implementation linear and nonlinear model predictive control algorithms. The computational burden has been reduced significantly by exploiting the structure of the MPC problem and the long prediction horizons used in Paper E would not have been practically implementable unless the MPC specific algorithms had been implemented.

No state estimator and disturbance models are used in the paper, as plant and control design model are identical in this simplified example and the states are assumed to be measurable.

Model predictive control, linear and nonlinear, enables knowledge of future disturbances, e.g. changes in wind speed, to be included in the prediction horizon

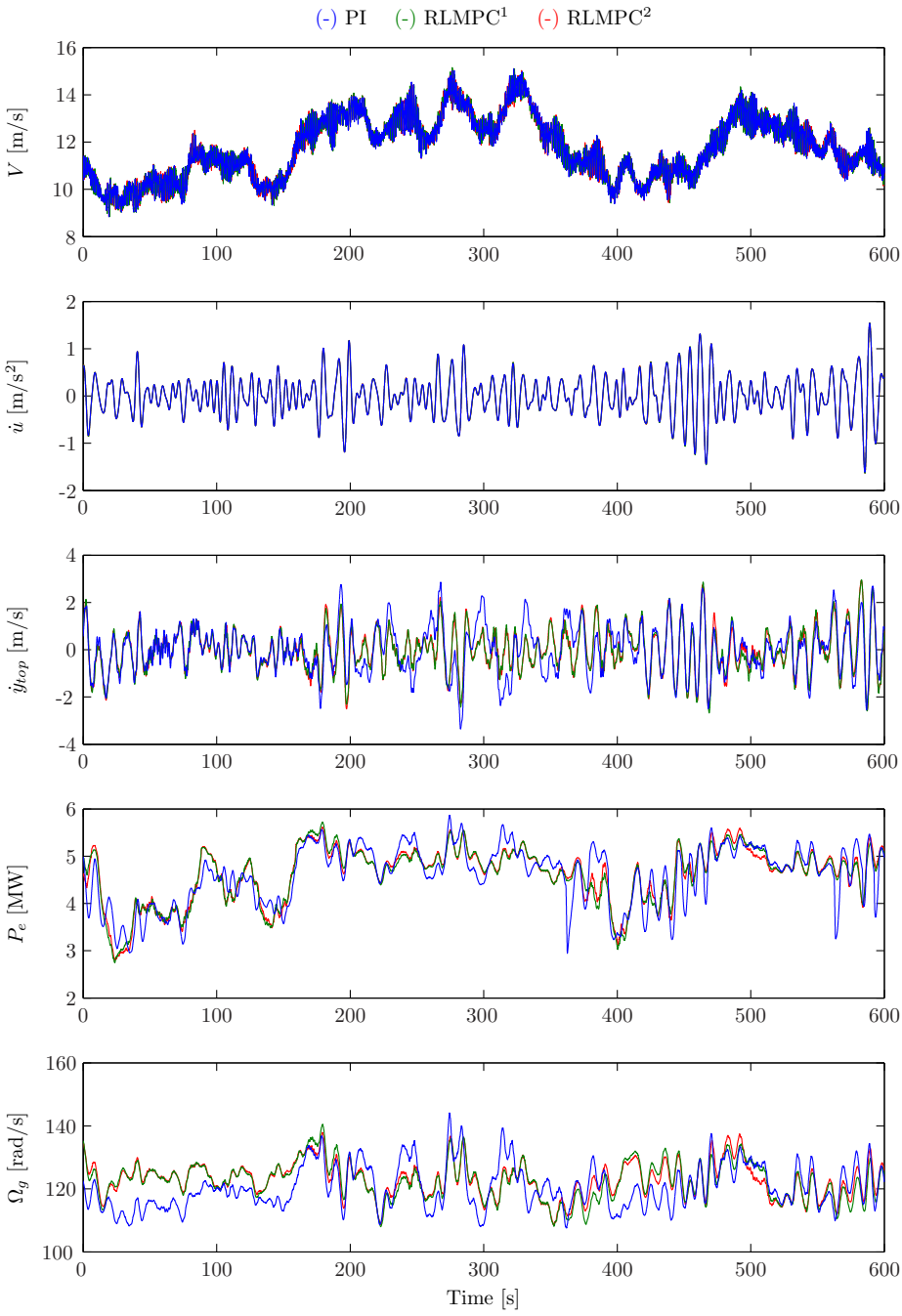
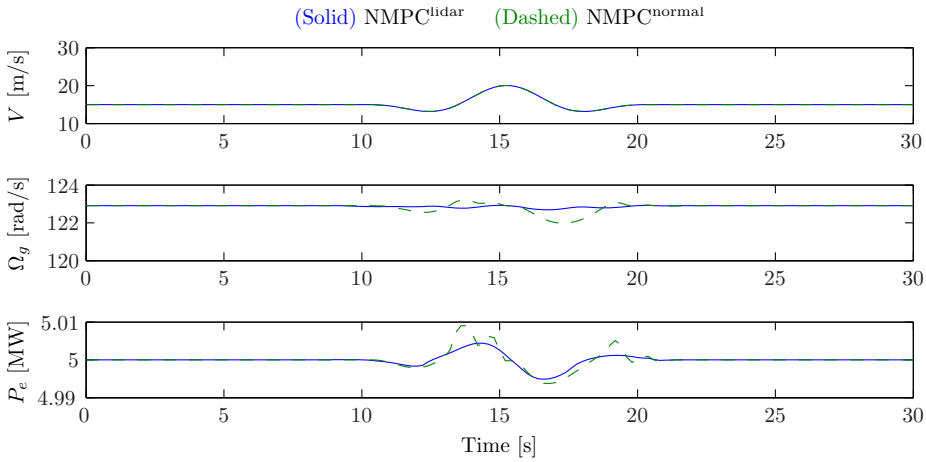


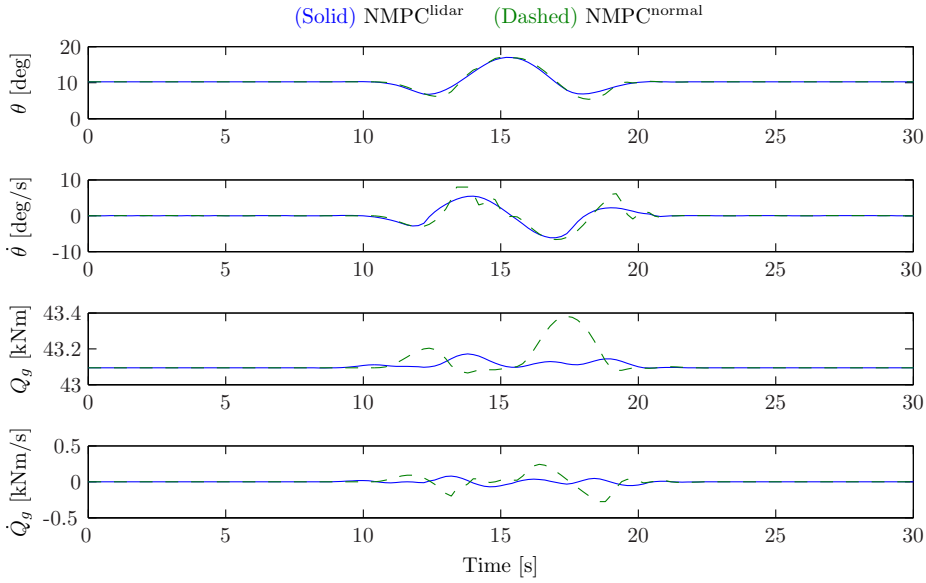
Figure 4.6: Wave forces (proportional to water acceleration $\dot{u}$) and turbulent wind with a mean speed of 12 m/s. Also depicted are tower top velocity, generator power and generator speed (Paper D).

of the model predictive controller. In Paper E two nonlinear model predictive controllers are compared, one with knowledge of future wind speeds obtained via a LIDAR and the other controller has no future knowledge of the wind speed and assumes it to be constant within the prediction horizon. Fig. 4.7 shows that the controller with future knowledge at hand is better equipped to reject the sudden change in wind speed caused by an extreme operating gust. The potential benefit of LIDARs has also been investigated by Santos [20] but a different implementation of the NMPC algorithm meant that Santos was unable to use long control horizons due to the computational burden of the NMPC problem and could thus not exploit the full potential NMPC.

Papers C and D use a modified primary control objective for partial load operation, by keeping the collective blade pitch constant control and tracking the optimal generator speed with generator torque control action. In paper E this is not needed, Figure 4.8 depicts the prediction horizon of the optimal computed sequence of control inputs based on the cost function with primary control objective given as (4.1a). The primary control objective dictates that the produced generator power should be driven towards the nominal power value. For below rated wind speed operation this cannot be achieved and as a consequence the generator power production is maximized to get as close to nominal power as possible. At the end of the prediction horizon this leads to undesirable behavior as generator power is increased at the expense of generator speed. The optimization is repeated in every sample and the undesirable behavior observed in the end of the prediction horizon is not observed in simulations as the end of the prediction horizon is pushed one sample further back in each sample, also known as receding horizon control. The prediction horizon should be long enough to ensure that the behavior observed in the end of the prediction horizon does not affect the behavior in the beginning of the prediction horizon. The need for a long prediction horizon in partial load operation along the with computational cost of solving the QP several times within the nonlinear optimization algorithm renders the NMPC controller slow and not suitable for real world application, unless dedicated hardware and optimized numerical libraries are used. This has however not been investigated in this work.



(a) Wind speed, generator speed and power.



(b) Controlled inputs to plant: Pitch angle and rate, and generator torque and rate.

Figure 4.7: Above rated wind speed: Simulations where an extreme operating gust occurs. Two nonlinear model predictive controllers are compared: The first, NMPC$^{\text{lidar}}$, includes knowledge of future wind speeds in the prediction horizon. The second, NMPC$^{\text{normal}}$, assumes that the current wind speed remains constant throughout the prediction horizon (Paper E).

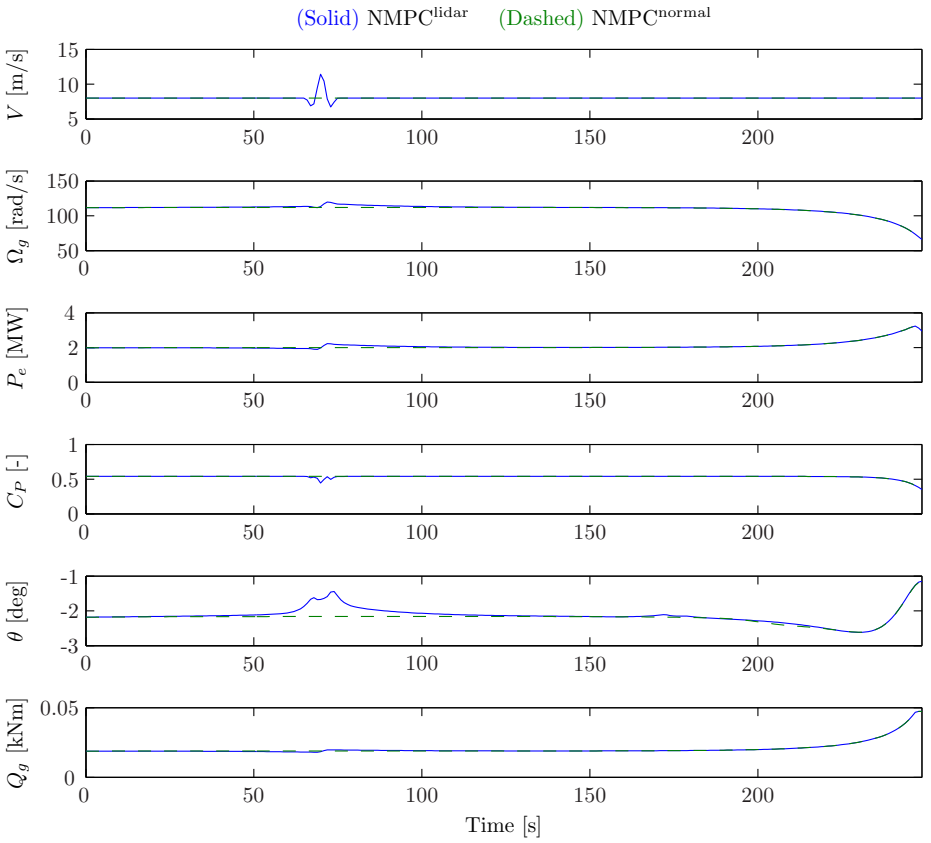


Figure 4.8: Partial load operation: Prediction horizon of NMPC with no terminal cost. The plots show the predictions made by the controller at a given point in time (Paper E).

Chapter 5

Conclusion and future development

The papers presented in this thesis are divided into three parts: Wind turbine modeling, wind turbine control and model predictive control implementational issues and this conclusion will be divided in a similar manner.

Under the naïve assumption that wind speed estimation is a measure for the potential fatigue load reduction of the model based controller Papers A and B have investigated which degrees of freedom that should be included in the control design model used by the model-based control and state estimation algorithms.

Paper A showed that dynamic inflow should be included the control design model to improve wind speed estimation. Paper B investigated a floating wind turbine and concluded that wave forces, tower bend and tower pitch are important degrees of freedom when estimating the wind speed for a floating wind turbine.

Papers C, D and E form a development towards having single control law for the entire wind speed range. The controllers of all three papers are shown to cover the entire wind speed range. The papers also form a development going from linear MPC to nonlinear MPC. The controllers in Papers C and D both use state estimators and are rely solely on realistic sensors, whereas the controller

in Paper E use a LIDAR as sensor, which currently is not a standard sensor.

Paper C, consisting of four linear model predictive controllers demonstrated the controllers ability to handle constraint on actuator states during a extreme operating gust, an event that can cause unstable behavior unless constraint handling of actuator position, rates and accelerations is implemented in the controller. The benchmark PI controller that was used to compare the performance of the MPC controller had to be modified from the standard configuration to handle such constraints. The original configuration of the PI controller entered an unstable limit-cycle-like behavior when constraints where met due to integrator-windup. Furthermore, switching conditions and other measures ensures a smooth switching was presented in Paper C giving a full wind speed range controller.

Paper D, consisting of single controller which can switch between partial and full load objectives showed how a floating wind turbine was controlled for the full wind speed range. Fatigue load reductions were observed when compared to a benchmark PI controller. Relinearization of the controller in each sample as well as reference tracking and disturbance rejection gathered in a single formulation was a novel development compared to the controller of Paper C simplifying the setup of the controller. The paper also investigated whether or not the inclusion dynamic inflow in the control design model has any significant impact on control performance. Results were not clear but suggested that the inclusion of dynamic inflow in the control design model is beneficial, especially around rated wind speed.

Paper E, representing the early steps in the implementation of a nonlinear model predictive controller, showed results for a simplified wind turbine. It was showed how the knowledge of future wind speed obtained with e.g. a LIDAR could be incorporated into the framework of NMPC and included in the prediction horizon to reject an incoming extreme operating gust. Fast algorithms to handle the computational burden posed by a NMPC algorithm have been implemented and presented in Papers F and G. The algorithms exploit the special structure of the NMPC problem to reduce the computational burden.

At present the MPC formulation of Paper D is the most promising because implementation of a real-time version should not be to difficult a task and the control design model relinearizes in each sample. The NMPC algorithm of Paper E should be implemented on dedicated hardware with hardware specific algorithms if real-time performance should be achieved.

The setup in Paper D is also prepared for individual pitch control, but further development of the control design model should be performed before individual blade pitch works in practice. In particular blade dynamics are expected to

be included in the control design model to achieve better individual blade pitch control. The inclusion of blades in the control design model would both improve wind speed estimation performance and give a better description of the blade root loads which are the sum of the aerodynamic forces and the kinematic forces of the dynamic blade structure. Fatigue loads of the tower side-side degree of freedom can not be targeted at present and the interaction between tower side-side movement and rotor and generator torque could be added to the control design model if tower side-side fatigue load reduction was to be targeted by the controller.

Bibliography

[1] Tony Burton, David Sharpe, Nick Jenkins, and Ervin Bossanyi. *Wind Energy Handbook*. John Wiley And Sons Ltd, 2001.

[2] Martin O. L. Hansen. *Aerodynamics of Wind Turbines*. James & James (Science Publishers) Ltd, second edition, 2008.

[3] Jason M. Jonkman and Marshall L. Buhl Jr. Fast user's guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, August 2005.

[4] P. Passon, M. Kuhn, S. Butterfield, J. Jonkman, T. Camp, and T.J. Larsen. Oc3-benchmark exercise of aero-elastic offshore wind turbine codes. *J. Phys., Conf. Ser. (UK)*, 75(1):–, 2007. ISSN 17426596.

[5] Torben Juul Larsen and Anders Melchior Hansen. How 2 hawc2, the user's manual. Technical Report Risø-R-1597(ver. 3-1)(EN), Risø National Laboratory, 2007.

[6] L.Y. Pao and K.E. Johnson. A tutorial on the dynamics and control of wind turbines and wind farms. *2009 American Control Conference (ACC-09)*, pages 2076–2089, 2009.

[7] J.H. Laks, L.Y. Pao, and A.D. Wright. Control of wind turbines: past, present, and future. *2009 American Control Conference (ACC-09)*, pages 2096–2103, 2009.

[8] W.E. Leithead and Bill Connor. Control of variable speed wind turbines: Design task. *Int J Control*, 73(13):1189–1212, 2000. ISSN 00207179.

[9] Morten H. Hansen, Anca Hansen, Torben J. Larsen, Stig Øye, Poul Sørensen, and Peter Fuglsang. Control design for a pitch-regulated,

variable speed wind turbine. Technical Report Risø-R-1500(EN), Risø National Laboratory, 2005.

[10] E.A. Bossanyi. Wind turbine control for load reduction. *Wind Energy*, 6 (3):229–244, 2003. ISSN 10954244.

[11] Torben Juul Larsen, Helge A. Madsen, and Kenneth Thomsen. Active load reduction using individual pitch, based on local blade flow measurements. *Wind Energy*, 8(1):67–80, 2005. ISSN 10954244.

[12] Alan D. Wright and Mark J. Balas. Design of state-space-based control algorithms for wind turbine speed regulation. *J Sol Energy Eng Trans ASME*, 125(4):386–395, 2003. ISSN 01996231.

[13] K. Selvam, S. Kanev, J. W. van Wingerden, T. van Engelen, and M. Verhaegen. Feedback–feedforward individual pitch control for wind turbine load reduction. *Int. J. Robust Nonlinear Control*, 19(1):72–91, 2009. ISSN 1099-1239. URL `http://dx.doi.org/10.1002/rnc.1324`.

[14] Karl A. Stol and Mark J. Balas. Periodic disturbance accommodating control for blade load mitigation in wind turbines. *J Sol Energy Eng Trans ASME*, 125(4):379–385, 2003. ISSN 01996231.

[15] F. D. Bianchi, R. J. Mantz, and C. F. Christiansen. Gain scheduling control of variable-speed wind energy conversion systems using quasi-lpv models. *Control Engineering Practice*, 13(2):247 – 255, 2005. ISSN 0967-0661. doi: DOI:10.1016/j.conengprac.2004.03.006. URL `http://www.sciencedirect.com/science/article/B6V2H-4C6KNNJ-2/2/aaa1c5d287142e3d3015f181f47a7ae6`.

[16] K.Z. Østergaard, J. Stoustrup, and P. Brath. Linear parameter varying control of wind turbines covering both partial load and full load conditions. *International Journal of Robust and Nonlinear Control*, 19(1):92–116, 2009. ISSN 10498923.

[17] A. Kumar and K. Stol. Simulating feedback linearization control of wind turbines using high-order models. *Wind Energ.*, 13(5):419–432, 2010. ISSN 1099-1824. URL `http://dx.doi.org/10.1002/we.363`.

[18] B. Boukhezzar, H. Siguerdidjane, and M. Maureen Hand. Nonlinear control of variable-speed wind turbines for generator torque limiting and power optimization. *Journal of Solar Energy Engineering*, 128(4):–, 2006. ISSN 01996231.

[19] Lorenzo Trainelli, Walter Sirchi, Barbara Savini, Alessandro Croce, and Carlo L. Bottasso. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody System Dynamics*, 16 (3):291–308, 2006. ISSN 13845640.

[20] R. Santos. *Damage mitigating control for wind turbines.* PhD thesis, University of Colorado at Boulder, United States – Colorado, 2007. URL `http://proquest.umi.com/pqdlink?did=1296104581&Fmt=7&clientId=79356&RQT=309&VName=PQD`.

[21] H. Snel and J.G. Schepers. Joint investigation of dynamic inflow effects and implementation of an engineering method. Technical Report ECN-C–94-107, ECN Wind Energy, 1995. URL `http://www.ecn.nl/publications/PdfFetch.aspx?nr=ECN-C--94-107`.

[22] T.G. van Engelen and E.L. van der Hooft. Dynamic inflow compensation for pitch controlled wind turbines. In *European Wind Energy Conference*, London, 2004.

[23] S. Øye. Unsteady wake effects caused by pitch -angle changes. In *IEA R&D WECS Joint Action on Aerodynamics of Wind Turbines, 1st Symposium*, pages 58–79, London, 15th October 1986.

[24] J. Jonkman. Dynamics modeling and loads analysis of an offshore floating wind turbine. Technical Report NREL/TP-500-41958, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401-3393, November 2007. URL `http://www.nrel.gov/docs/fy08osti/41958.pdf`.

[25] Bjorn Skaare, Tor David Hanson, and Finn Gunnar Nielsen. Importance of control strategies on fatigue life of floating wind turbines. *ASME Conference Proceedings*, 2007(42711):493–500, 2007. doi: 10.1115/OMAE2007-29277. URL `http://link.aip.org/link/abstract/ASMECP/v2007/i42711/p493/s1`.

[26] C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of interior-point methods to model predictive control. *J. Optim. Theory Appl. (USA)*, 99 (3):723–757, 1998. ISSN 00223239.

[27] Matthew J. Tenny, Stephen J. Wright, and James B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, 2004. ISSN 15732894.

[28] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, 2nd edition, 2006.

[29] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004. URL `http://www.stanford.edu/~boyd/cvxbook/`.

[30] Gabriele Pannocchia, Nabil Laachi, and James B. Rawlings. A candidate to replace pid control: Siso-constrained lq control. *AIChE Journal*, 51(4): 1178–1189, 2005. ISSN 00011541.

[31] Gabriele Pannocchia and James B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2):426–437, 2003. ISSN 00011541.

[32] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *Automatic Control, IEEE Transactions on*, 43(8):1163 –1169, 1998. ISSN 00189286.

Part I

Wind Turbine Modeling

WIND ENERGY
Wind Energ. 2010; **00**:1–11
DOI: 10.1002/we

RESEARCH ARTICLE

The effect of dynamic inflow in free mean wind speed estimation

L. C. Henriksen [1], M. H. Hansen [1], N. K. Poulsen [2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark
[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ABSTRACT

Model-based state space controllers require knowlegde of states, both measureable and unmeasureable, and state estimation algorithms are typically employed to obtain estimates of the unmeasureable states. For the control of wind turbines, a good estimate of the free mean wind speed is important for the closed-loop dynamics of the system and an appropiate level of modeling detail is required to obtain good estimates of the free mean wind speed. In this work three aerodynamic models based on blade element momentum theory are presented and compared to the aero-servo-elastic code HAWC2. The first model known as quasi-steady aerodynamics assumes instant equilibrium of the wind turbine wake, a modeling approach often used by model-based control algorithms. The second model includes the dynamic wake also known as dynamic inflow and gives a more correct description of the actual physics of the wind turbine wake. The dynamic inflow model includes a number of dynamic states proportional to the number of radial points in the spatially discretized blade formulation. The large number of dynamic states inhibits the use of this model in model-based control and estimation algorithms. The lack of dynamic inflow the first model and large number of dynamic states in the second model leads to a third model. A simplified dynamic inflow model, which with only a single dynamic state is still able to capture the most significant dynamics of the more advanced dynamic inflow model. Simulations in the aero-servo-elastic code HAWC2 compare the ability to estimate the free mean wind speed when either the first or third model is included in the estimation algorithm. The degrees of freedom for the wind turbine used in the simulations, have been reduced to achieve more transparent results and the simulations clearly show that the inclusion of the dynamic inflow model in the free wind speed estimation algorithm is important for good free mean wind speed estimates. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS

orizontal axis wind turbine; dynamic inflow; blade element momentum theory; extended kalman filter

Correspondence
L. C. Henriksen, Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark.

E-mail: larh@risoe.dtu.dk

Received . . .

1. INTRODUCTION

Dynamic wake or dynamic inflow describes the lag with which the wake induced by a wind turbine settles to a new equilibrium when operating conditions of the wind turbine are changed, e.g. a sudden change in pitch angle. Dynamic inflow has long been regocnized as a significant phenomonon which should be taken into account when modelling wind turbines in e.g. aero-servo-elastic codes [1, 2, 3]. The importance of dynamic inflow is not new knowlegde for the wind energy community in general as the cited works are dated to the late 1980s and early 1990s. It has already been established that taking the dynamic inflow into account when designing a controller leads to reduced loads [4] and classic proptional-integral controllers have been tuned to achieve certain closed-loop properties by taking the dynamic inflow into account [5]. But for model-based control and state estimation methods applied to wind turbines the quasi-steady power and thrust coffecients, usualy denoted C_P and C_T, are most often used [6, 7].

Prepared using weauth.cls [Version: 2010/06/17 v1.00]

Model-based control and state estimation require a simple aerodynamic model which is usually based on blade element momentum theory (BEM). In reality temporal dynamics on several time scales occur, giving rise to dynamic inflow on the slow end of the time scale and dynamic stall on the fast end of the time scale. In this work a simple dynamic inflow model inspired by the model proposed by Ref. [1] and also described in Ref. [8] and Ref. [3] has led to the development of an even simpler dynamic inflow model suited for model-based control and state estimation purposes. The two dynamic inflow models along with a quasi-steady aerodynamic model are compared to aerodynamic model implemented in the aero-servo-elastic code HAWC2 [9, 10]. Many degrees of freedom have been disabled in the simulations to enable a clearer visualization of the effects of dynamic inflow.

If model-based controllers are to take dynamic inflow into account, they are dependant on state estimation methods able to estimate the states of the dynamic inflow model, e.g. the free mean wind speed and induced velocities. In the present work estimations based on extended Kalman filter theory compare the ability to estimate the free mean wind speed and the induced velocities when using either the developed dynamic inflow model or the quasi-steady aerodynamic model in the extended Kalman filter. Results are clear and show that dynamic inflow should be taken into account, if the free mean wind speed and induced velocities are to be estimated correctly during changes in operating conditions for the wind turbine.

This paper is structured as follows: In the first section blade element momentum theory is revisited to form the basis of the next section which describes the quasi-steady and and dynamic inflow models. After that a section presenting comparisons between the different BEM models and HAWC2. Finally the results for the free wind speed estimation are presented and discussed and conclusions are made.

2. BLADE ELEMENT MOMENTUM THEORY

Blade element theory describes the aerodynamic forces at a finite number of blade elements along the blade span. The aerodynamic forces cause a blockage of the air flow and induce an air flow in the oposite direction of the free wind speed. Fig. 1 gives an overview of the local forces and flow velocities in a single blade element. Momentum theory is used to determine the equilibrial conditions for the free and induced wind speeds. In this section basic blade element momentum theory (BEM) is outlined for the blade elements and in the last part of the section the aerodynamic forces along the blade span are integrated to give rotor-wide forces.

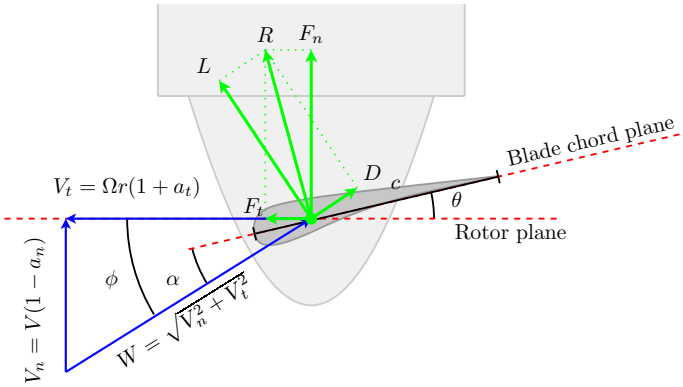


Figure 1. Cross section of blade in the span wise direction along the blade.

The induced velocities tangential v_t and normal v_n to the rotor plane are described by the induction factors a_t and a_n, the resulting inflow velocity in the normal and tangential directions is

$$V_n = V - v_n = V(1 - a_n),\ v_n = Va_n \tag{1}$$

$$V_t = V_r + v_t = \Omega r(1 + a_t),\ v_t = \Omega r a_t \tag{2}$$

The absolute value of the inflow velocity seen by the blade and the inflow angle are

$$W = \sqrt{V_n^2 + V_t^2} \tag{3a}$$

$$\phi = \arctan \epsilon, \quad \epsilon = \frac{V_n}{V_t} \tag{3b}$$

Wind Energ. 2010; **00**:1–11 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

The angle of attack is the difference between the inflow angle and the pitch of the blade element

$$\alpha = \phi - \theta \tag{4}$$

The inflow triangle displayed in Fig. 1 gives the aerodynamic forces where drag D is parallel to the inflow velocity and lift L is normal to the inflow velocity

$$L = \frac{1}{2}\rho W^2 C_L(\alpha)c \tag{5}$$

$$D = \frac{1}{2}\rho W^2 C_D(\alpha)c \tag{6}$$

Lift and drag can be projected into forces tangential F_t and normal F_n to the rotor plane

$$F_n(r) = L\cos\phi + D\sin\phi \tag{7}$$

$$F_t(r) = L\sin\phi - D\cos\phi \tag{8}$$

The aerodynamic torque Q_i and thrust T_i for each blade multiplied with the number of blades n_B gives the rotor-wide aerodynamic torque and thrust

$$Q = n_B \int_0^R r F_t(r)\, dr \tag{9}$$

$$T = n_B \int_0^R F_n(r)\, dr \tag{10}$$

Quasi-steady values for the induction factors along the blade span a_n^{qs} and a_t^{qs} can be computed with the BEM algorithm given in [8] where Prandtl's tiploss correction and Glaurt's axial induction correction are also used. There are several versions of Glaurt's axial induction correction for high induction factors and depending on which model is used slightly different quasi-steady axial induction factors are found. The axial indution factor correction for high axial induction factors used in this work is the same as the one used in HAWC2 [11].

In steady state the aerodynamic coefficients C_P and C_T can be determined and are given by

$$C_P(\lambda,\theta) = \frac{n_B \Omega Q}{\frac{1}{2}\rho\pi R^2 V^3} \tag{11}$$

$$C_T(\lambda,\theta) = \frac{n_B T}{\frac{1}{2}\rho\pi R^2 V^2} \tag{12}$$

where the tip-speed-ratio

$$\lambda = \frac{\Omega R}{V} \tag{13}$$

enables a two-dimensional (λ,θ) decription of the steady state aerodynamic forces rather than a three-dimensional (V,Ω,θ).

3. AERODYNAMIC MODELS

In the following three different aerodynamic models are presented: The quasi-steady model (BEM$^{\text{qs}}$) assumes instant equilibrium of the wake when conditions change. The full state dynamic inflow model (BEM$^{\text{full}}$) includes as many tangential and normal induced wind speeds as there are radial points in the spatial discretized blade description. The simplified dynamic inflow model (BEM$^{\text{simple}}$) assumes that the tangential induced wind speeds are quasi-steady and the normal induced wind speeds are described by one average axial induction wind speed as will be explained in the following.

3.1. Quasi-steady aerodynamic model

Under the assumption of a quasi-steady wake the induced velocities are assumed to settle to their stationary values instantly. This means that the induced velocities are

$$v_n(r) = V a_n^{qs}(\lambda,\theta,r) \tag{14}$$

$$v_t(r) = \Omega r a_t^{qs}(\lambda,\theta,r) \tag{15}$$

These induced velocities can be inserted in the BEM equations of the previous section to compute the aerodynamic forces. A computationally faster approach is to calculate the aerodynamic forces via precalculated look-up tables for the power and thrust coefficents C_P and C_T

$$Q = \frac{\frac{1}{2}\rho\pi R^2 V^3 C_P(\lambda,\theta)}{\Omega} \tag{16}$$

$$T = \frac{1}{2}\rho\pi R^2 V^2 C_T(\lambda,\theta) \tag{17}$$

an approach typically used by model-based control and state estimation algorithms.

3.2. Full state dynamic inflow aerodynamic model

The dynamic inflow describes the slow time scale of the aerodynamics of a wind turbine. The model captures the behavior of the wake which needs to settle to a new equilibrium before the aerodynamic forces have settled to their new steady state values whenever changes in operating conditions occur. Time constants of the dynamic inflow are in the order rotor radius divided by free wind speed. Different dynamic inflow models have slightly different formulations of the time constants and filter configurations [3] . The dynamic inflow model chosen in this work is a simplfied version of one orginally suggested by Ref. [1] and documented in several later works [3, 8] . The orignal model consist of a first order filter with a time constant determined by rotor-wide considerations. The output of this filter is fed into a filter with a radial dependant time constant. In this work only the first filter is included in the model

$$\tau_1 \dot{v}_n(r) + v_n(r) = V a_n^{qs}(\lambda,\theta,r) \tag{18}$$

$$\tau_1 \dot{v}_t(r) + v_t(r) = r\Omega a_t^{qs}(\lambda,\theta,r) \tag{19}$$

where the time constant is

$$\tau_1^{\emptyset ye} = \frac{1.1R}{V - 1.3\bar{v}_n}, \quad \bar{v}_n = \frac{1}{R}\int_0^R v_n(r)dr \tag{20}$$

and the radial dependant temporal dynamics have been omitted from this work. In the original model, the input to the filter includes, apart from the quasi-steady induced inflow velocities, also a term with the rate of change of the quasi-steady induced inflow velocities. This term has also been omitted from the BEM$^{\text{full}}$ and comparisons between the models in the next section has resulted in an adjusted time constant

$$\tau_1 = \frac{1}{2}\tau_1^{\emptyset ye} \tag{21}$$

which as a results has been used in both the full and simplified dynamic inflow models throughout this work.

3.3. Simplified dynamic inflow aerodynamic model

The full state dynamic inflow model can be simplified and reduced to having only a single state at the expense of the modeling detail level. This simplication can be justied by the fact that it will result in a pratical model which is implementable in model-based control and state estimation algorithms. The tangential induced wind speeds are assumed to be quasi-steady as in the BEM$^{\text{qs}}$ model and the induced axial wind speed states of the BEM$^{\text{full}}$ model are gathered in a single averaged induced axial wind speed state with the temporal dynamics

$$\tau_1 \dot{\bar{v}}_n + \bar{v}_n = V \bar{a}_n^{qs}(\lambda,\theta) \tag{22}$$

with the same time constant as in the previous subsection

$$\tau_1 = \frac{1}{2}\frac{1.1R}{V - 1.3\bar{v}_n} \tag{23}$$

Notice that $\bar{v}_n$ is a function of the states $v_n(r_1),\ldots,v_n(r_{nr})$ in the BEM$^{\text{full}}$, whereas in the BEM$^{\text{simple}}$ model $v_n(r_1),\ldots,v_n(r_{nr})$ are outputs dependant of the state $\bar{v}_n$. In the BEM$^{\text{simple}}$, the radial dependant induced axial wind speeds are distributed along the blade span with the quasi-steady distribution

$$v_n(r) = \frac{a_n^{qs}(\lambda,\theta,r)}{\bar{a}_n^{qs}(\lambda,\theta)}\bar{v}_n, \quad \bar{a}_n^{qs}(\lambda,\theta) > 0 \tag{24}$$

Wind Energ. 2010; **00**:1–11 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

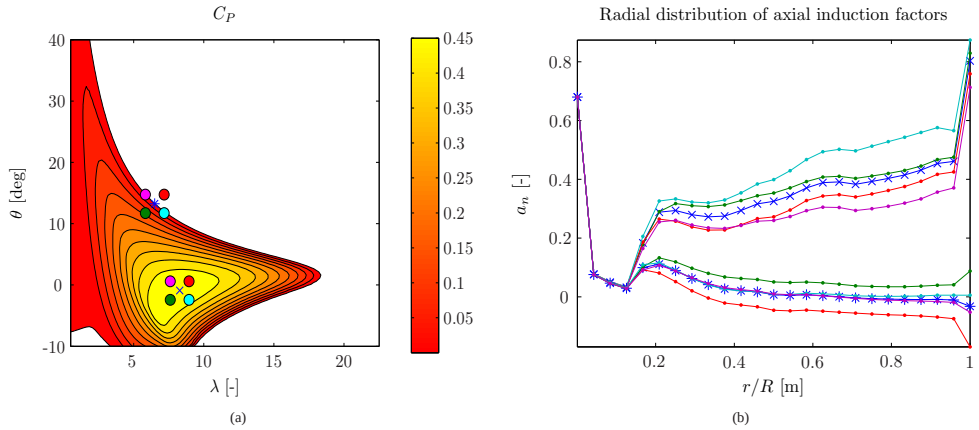


Figure 2. Quasi-steady power coefficent curve - Fig. 2(a) Shows the quasi-steady power coefficient curve, two points of interest, each with four neighbouring points are plotted on the figure. The radial distribution of the axial induction factors for the two points and their neighbouring points are shown in Fig. 2(b).

where the average axial induction factor can be calculated offline

$$\bar{a}_n^{qs}(\lambda,\theta) = \frac{1}{R}\int_0^R a_n^{qs}(\lambda,\theta,r)dr \tag{25}$$

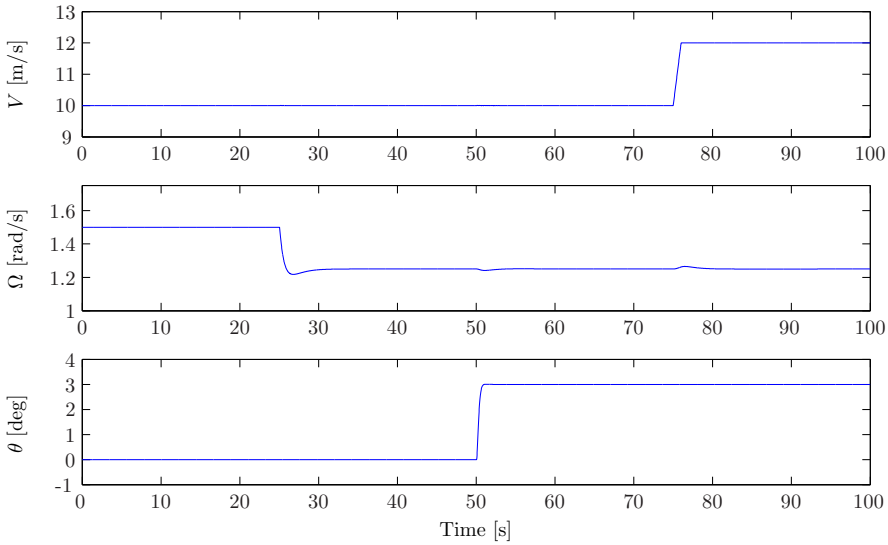
This assumption is valid if the dynamic and quasi-steady distributions along the blade are not too different. Figure 2 shows the quasi-steady distribution of axial induction factors for two main points each with four neighbouring points and here it is seen that the distributions for neighbouring points are quite similar, as will be demonstrated in the next section where the different models are compared with each other and with the more complex model implemented in HAWC2.

4. COMPARISON OF AERODYNAMIC MODELS

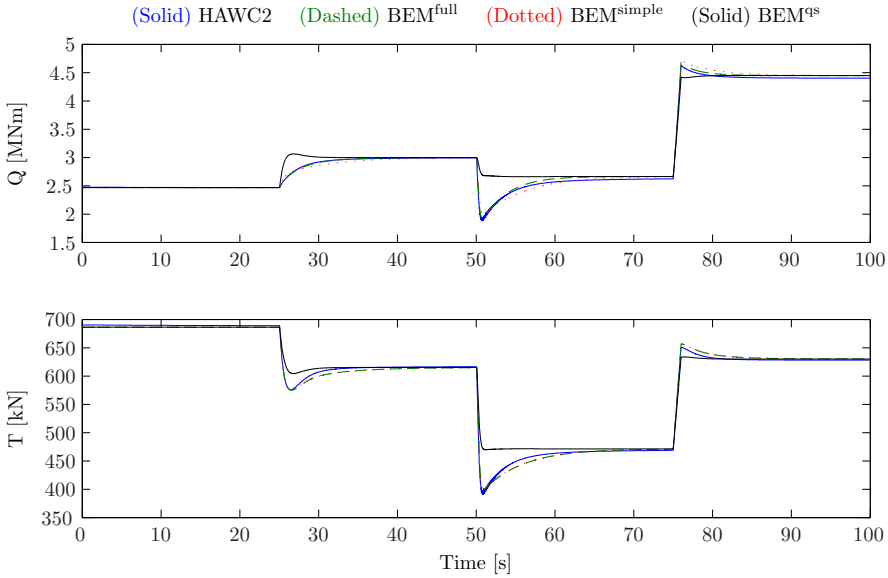
In this section the different BEM models presented in the previous section are compared to each other and to the aero-servo-elastic code HAWC2 [9] . The wind turbine used in this work is based on the NREL 5MW reference wind turbine [12] .

Simulations have been run where the rotor speed is controlled by a PI controller that via the generator torque controls the rotor speed towards its chosen setpoint. At time 25 s the rotor speed setpoint is changed and the PI controller regulates the rotor speed to its new setpoint. At time 50 s the pitch angle is increased and at time 75 s the wind speed is ramped to a higher wind speed. These changes of operating conditions happen quite fast to better visualize the differences between the presented aerodynamic models. The PI controller used in this section is not to be confused with the controller used for normal operation of the wind turbine. The sole puporse of this sections PI controller is to control the generator speed via the generator torque.

A simulation, around rated wind speed operational conditions, has been run where the wind turbine has been simplified by stiffening the structural properties of the blades, tower and drive-shaft. Furthermore, rotor tilt, coning and dynamic stall have all been disabled. Tower shadow and wind shear also have been disabled. All these simplifications are done to make comparison between the BEM code and HAWC2 easier. The outcome is shown in Fig. 3 where it is clear that the BEM^{full} and $\text{BEM}^{\text{simple}}$ capture the dynamics of the HAWC2 output far better than the BEM^{qs} model. Although the outputs of HAWC2, BEM^{full} and $\text{BEM}^{\text{simple}}$ are not identical they are quite similar and it is concluded that the $\text{BEM}^{\text{simple}}$ is suitable for model-based control design and state estimation purposes.



(a) Inputs to aerodynamic models



(b) Outputs from aerodynamic models

Figure 3. Simulations in HAWC2 with stiffened turbine with no wind shear, tower shadow etc. Fig. 3(b) shows that the variuos BEM codes model the behavior of HAWC2

Wind Energ. 2010; **00**:1–11 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

5. ESTIMATION OF THE WIND SPEED

In this section an extended Kalman filter (EKF) [13] will be used to estimate the mean wind speed affecting the rotor. The performance of two EKFs based on two different models, namely the BEM$^{\text{simple}}$ and BEM$^{\text{qs}}$, will be compared to investigate the effect of dynamic inflow in wind speed estimation.

5.1. Extended Kalman Filter

Both models used by the EKFs share the rotor acceleration model

$$\dot{\Omega} = \frac{1}{J_t}(Q - Q_g), \quad J_t = J_r + N_g^2 J_g \tag{26}$$

where the total inertial mass of the wind turbine is the sum of the rotor inertial mass J_r and the generator inertial mass J_g felt through a gear with a gear ratio N_g. Both models used by the EKFs also share the free mean wind speed model, which is assumed to be a random walk process driven by zero-mean white noise

$$\dot{V} = w, \quad w \in \mathcal{N}(0, \sigma_w^2) \tag{27}$$

The models differ in the aerodynamic modelling where one model assumes quasi-steady aerodynamic forces and the other assumes dynamic inflow with temporal dynamics for the mean axial induced velocity given by eq. (22).

The state space ordinary differential equation for the model assuming quasi-steady aerodynamic is

$$\begin{bmatrix} \dot{\Omega} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_t}\left(Q(V, \Omega, \theta) - Q_g\right) \\ 0 \end{bmatrix} + \mathbf{G}w \tag{28}$$

where the process noise input matrix is $\mathbf{G} = [0\ 1]^T$. The state space ordinary differential equation for the model containing the simplified dynamic inflow model is

$$\begin{bmatrix} \dot{\Omega} \\ \dot{V} \\ \dot{\bar{v}}_n \end{bmatrix} = \begin{bmatrix} \frac{1}{J_t}\left(Q(V, \Omega, \theta, \bar{v}_n) - Q_g\right) \\ 0 \\ \frac{1}{\tau_1(V, \bar{v}_n)}\left(V\bar{a}_n^{qs}(V, \Omega, \theta) - \bar{v}_n\right) \end{bmatrix} + \mathbf{G}w \tag{29}$$

where the process noise input matrix is $\mathbf{G} = [0\ 1\ 0]^T$. The inputs to both models used by the EKFs are generator torque, denoted Q_g, and collective pitch angle, and the measurements available to the EKFs are rotor speed and acceleration

$$\boldsymbol{u} = [\theta\ Q_g]^T \text{ and } \boldsymbol{y} = [\Omega\ \dot{\Omega}]^T$$

the measurement ouput state space equation for the quasi-steady aredynamic models is

$$\begin{bmatrix} \Omega \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \Omega \\ \frac{1}{J_t}\left(Q(V, \Omega, \theta) - Q_g\right) \end{bmatrix} \tag{30}$$

and for the dynamic inflow model

$$\begin{bmatrix} \Omega \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \Omega \\ \frac{1}{J_t}\left(Q(V, \Omega, \theta, \bar{v}_n) - Q_g\right) \end{bmatrix} \tag{31}$$

Another output vector containing the variables which cannot be measured, but are used to compare the performance of the EKFs is

$$\boldsymbol{z} = [V\ \bar{v}_n\ Q\ T]^T$$

which has the state space output equation for the quasi-steady aerodynamic model

$$\begin{bmatrix} V \\ \bar{v}_n \\ Q \\ T \end{bmatrix} = \begin{bmatrix} V \\ V\bar{a}_n^{qs}(V, \Omega, \theta) \\ Q(V, \Omega, \theta) \\ T(V, \Omega, \theta) \end{bmatrix} \tag{32}$$

and for the dynamic inflow model

$$\begin{bmatrix} V \\ \bar{v}_n \\ Q \\ T \end{bmatrix} = \begin{bmatrix} V \\ \bar{v}_n \\ Q(V, \Omega, \theta, \bar{v}_n) \\ T(V, \Omega, \theta, \bar{v}_n) \end{bmatrix} \tag{33}$$

The states for the BEM$^{\text{qs}}$ model are

$$\boldsymbol{x} = [\Omega \; V]^T$$

and the states for the BEM$^{\text{simple}}$ model are

$$\boldsymbol{x} = [\Omega \; V \; \bar{v}_n]^T$$

The equations given above, constituting the model of the wind turbine, can be formulated in more general terms as a state space ordinary differential equation $\boldsymbol{f}$ and an output equation $\boldsymbol{g}$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t) \tag{34a}$$
$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{v}(t) \tag{34b}$$

where $\boldsymbol{w}$ is zero-mean white process noise with the covariance matrix $\mathbf{R}_{\mathbf{x}}$ and $\boldsymbol{v}$ is zero-mean white measurement noise with the covariance matrix $\mathbf{R}_{\mathbf{y}}$. From (34a) the state progress equation $\underline{\boldsymbol{f}}$, describing the progress of states from the discrete point in time t_k to t_{k+1},

$$\boldsymbol{x}(t_{k+1}) = \underline{\boldsymbol{f}}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)) + \boldsymbol{w}(t_k) \tag{35}$$

can be determined using e.g. Runge-Kutta time integration schemes. In the remainder of this paper, the time-discrete temporal argument $\boldsymbol{x}(t_k)$ will be shortened to $\boldsymbol{x}_k$ to ease notation. The total process noise of the system $\boldsymbol{w}$ is the sum of the process noise of the random walk wind model $\mathbf{G}w$ and a zero-mean white noise vector $\boldsymbol{w}_{\boldsymbol{x}}$ with the covariance matrix $r_x\mathbf{I}$. For both the EKF based on the quasi-steady aerodynamic model and for the EKF based on the simplified dynamic inflow model the total process noise covariance matrix has the structure

$$\mathbf{R}_{\mathbf{x}} = \mathbf{W}^T[\mathbf{G}^T\sigma_w^2\mathbf{G} + r_x\mathbf{I}]\mathbf{W} \tag{36}$$

where the $r_x\mathbf{I}$ term indicates the overall confidence in how well the state progress equation predicts the future states of the plant. A high r_x indicates high uncertainty due to stochastic disturbances or a mismatch between the model and plant. The scalar r_x has been chosen to be 10^{-4} suggesting that the model is to be trusted. The variance of the random walk wind model σ_w^2 has been chosen to be 1 suggesting that the free wind speed wind can change quite a lot from one time-discrete point to the next. The covariance matrix is also scaled with the scaling matrix $\mathbf{W}$ which is diag($[\frac{1}{\Omega_{nom}} \; \frac{1}{V_{nom}}]$) for the quasi-steady aerodynamic model and diag($[\frac{1}{\Omega_{nom}} \; \frac{1}{V_{nom}} \; \frac{1}{\bar{v}_{n,nom}}]$) for the dynamic inflow model and the nominal values are Ω_{nom} = 1.267 rad/s, V_{nom} = 25 m/s and $\bar{v}_{n,nom}$ = 10 m/s. The measurement noise covariance matrix displays high confidence in the measurements

$$\mathbf{R}_{\mathbf{y}} = \text{diag}\left(\begin{bmatrix} \frac{10^{-4}}{\Omega_{nom}^2} & \frac{10^{-4}}{\Omega_{nom}^2} \end{bmatrix}\right) \tag{37}$$

as no real measurement noise is assumed in this simplified experiment. With the formulation of state progress and measurement equations in place, the extended Kalman filter can be constructed. The estimated states are denoted $\hat{\boldsymbol{x}}_{k|k-1}$, where $k|k-1$ means the estimation at time k given by the information at time $k-1$. The *a-posteriori* estimation of the states is given by

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{L}_k[\boldsymbol{y}_k - \boldsymbol{g}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k)] \tag{38}$$

Enabling an *a-priori* estimation of the one-step-ahead prediction of the states

$$\hat{\boldsymbol{x}}_{k+1|k} = \underline{\boldsymbol{f}}(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k) \tag{39}$$

where the Kalman gain $\mathbf{L}_k$ and output error covariance $\boldsymbol{\Psi}_k$

$$\mathbf{L}_k = \mathbf{P}_{k|k-1}\mathbf{C}_{k|k-1}^T\boldsymbol{\Psi}_k^{-1} \tag{40a}$$
$$\boldsymbol{\Psi}_k = \mathbf{C}_{k|k-1}\mathbf{P}_{k|k-1}\mathbf{C}_{k|k-1}^T + \mathbf{R}_{\mathbf{y}} \tag{40b}$$

Wind Energ. 2010; **00**:1–11 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

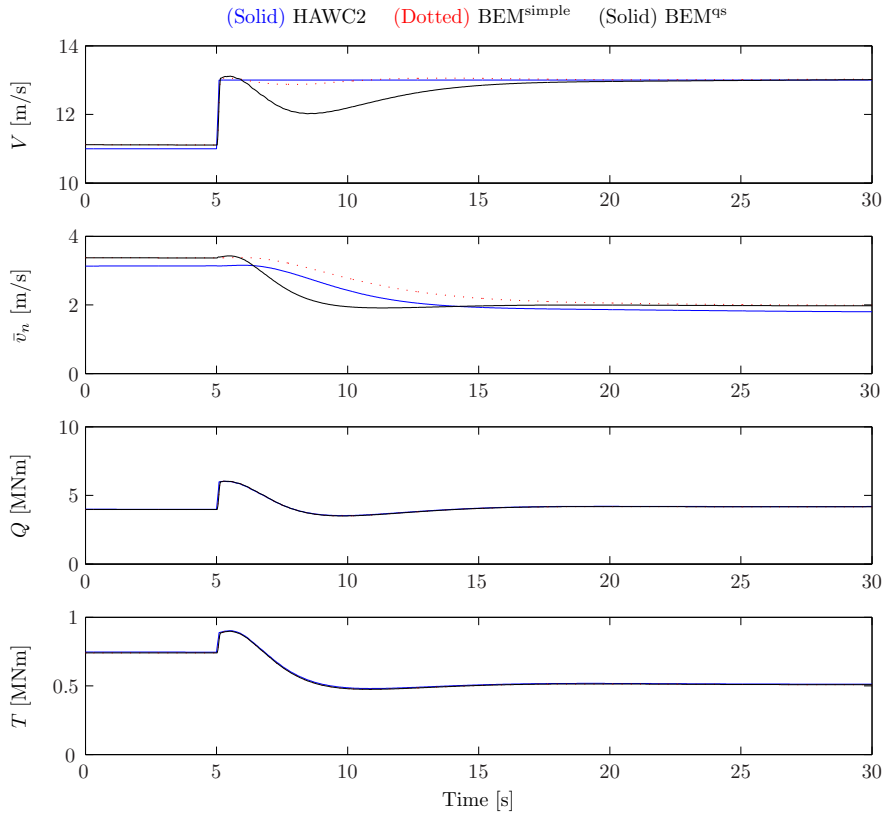


Figure 4. Estimation of free mean wind speed and averaged axial induced velocity and aerodynamic forces. The extended Kalman filter based on the BEM^{qs} model fails to predict the wind speed transient seen from time 5 s to 20 s, whereas the extended Kalman filter based on the BEM^{simple} model, which takes the dynamic inflow into account, captures the behavior.

are updated by the discrete time recursive Riccati equation

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k \mathbf{C}_{k|k-1} \mathbf{P}_{k|k-1} \tag{40c}$$

$$\mathbf{P}_{k+1|k} = \underline{\mathbf{A}}_{k|k} \mathbf{P}_{k|k} \underline{\mathbf{A}}_{k|k}^T + \mathbf{R}_{\mathbf{x}} \tag{40d}$$

The EKF is based on an extension of the Kalman filter, which is the optimal estimator for linear systems. Consequently the time discrete model consisting of (35) and (34b) is linearized w.r.t. $\boldsymbol{x}$ using first order Taylor series approximation giving $\underline{\mathbf{A}}_{k|k}$, which is the Jacobian of the state progress equation linearized at $(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k)$ and $\mathbf{C}_{k|k-1}$, which is the Jacobian of the output equation linearized at $(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k)$.

5.2. Results of the extended Kalman filter

A new simulation is executed where the stiffened turbine from the previous section is controlled around rated wind speed by a PI controller similar to the controller by Jonkman [12]. The performance of the two EKFs can be seen in Fig. 4 based on the inputs and outputs seen in Fig. 5. At time 5 s the wind speed changes from below rated wind speed to above rated wind speed causing the controller to react and pitch the blades.

From Fig. 4 it can be seen that both EKFs are able to capture the aerodynamic forces without being able to measure them, but the EKFs disagree about which wind speeds are causing the aerodynamic forces to be as they are. The key to understand how both EKFs can predict similar aerodynamic forces but dissimilar wind speeds is in the difference between the two aerodynamic models used by the EKFs.

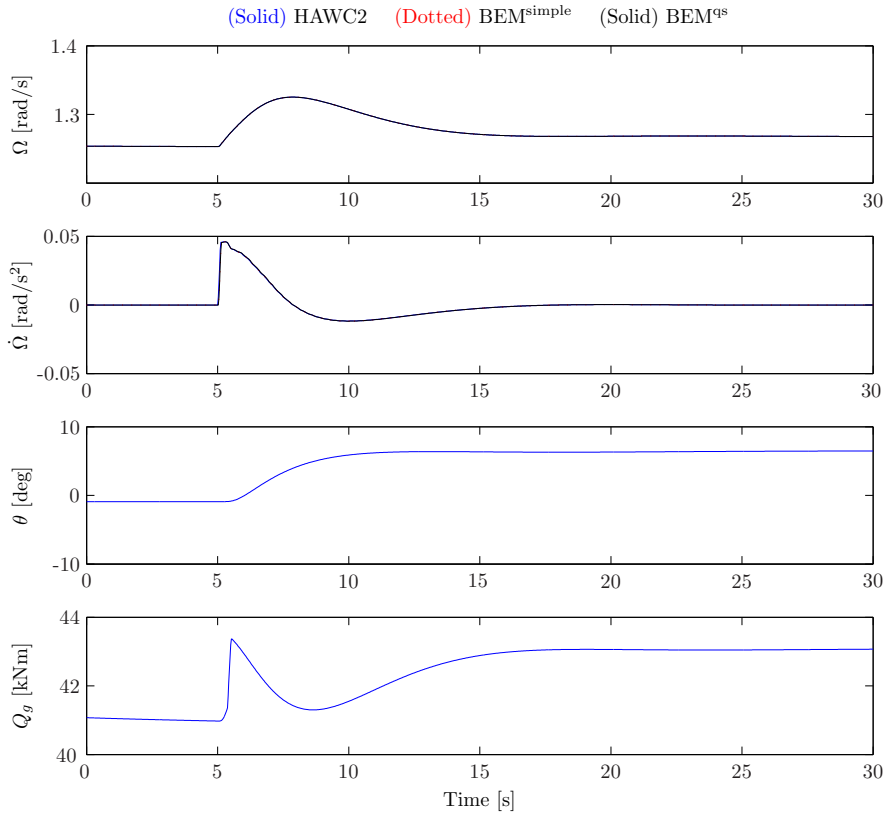


Figure 5. The inputs and measurements available to the two extended Kalman filters.

First it has to be explained that with the available sensors, rotor speed and acceleration, the Kalman filters have no direct information about the wind speed. From the rotor acceleration equation (26) it can be seen that with the knowlegde of generator torque and rotor acceleration, the aerodynamic torque can be determined correctly regardless of the aerodynamic model. Because the aerodynamic torque and thrust are both functions of the same inflow triangle, seen in Fig. 1, a correct estimation of the aerodynamic torque leads to correct estimations of the inflow conditions (3), which in turn leads to correct estimation of the aerodynamic thrust for both aerodynamic models even though no measurements give any direct information about the aerodynamic thrust.

It remains to be determined why the wind speed estimates of the two Kalman filters are different. The wind speed estimate of the EKF based on the dynamic inflow model is almost correct, because the $\text{BEM}^{\text{simple}}$, as established in the previous section, provides a good describtion of the aerodynamic model implemented in HAWC2. The wind speed estimate based on the quasi-steady aerodynamic model is not as good because the aerodynamic model only resembles that of HAWC2 during steady state. It can be seen from the Figures 4 and 5 that the transient of the wind speed estimate for the quasi-steady model is highly correlated with the transient of the measured rotor speed and acceleration as that is the only information available to the quasi-steady aerodynamic based EKF.

It could seem puzzling that the EKF based on the quasi-steady aerodynamic model is able to capture the steep gradient of the wind speed at time 5 s and that the estimated induced mean axial velocity does not change instanly to the new equilibrium perscribed by the quasi-steady aerodynamic model. It should be understood that an estimation based on a particular model is not the same as a simulation based on the same model. The estimation algorithm will at each sample update its estimates based on measurements from the plant, so it is the physics of the plant and measurement updates thereof, which are driving the estimates and not the actual model used by the estimation algorithm. The model used by the estimation algorithm is simply responsible for how well the measurements are interpreted and used to form estimates

Wind Energ. 2010; **00**:1–11 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

of the states. Which is why the EKF based on the quasi-steady aerodynamic model is unable to estimate the wind speed correct during the transient seen from 5 s to 20 s.

From the results presented in Figure 4 it has been demonstrated that to estimate the free mean wind speed a dynamic inflow model should be included in the model-based state estimation algorithm as transients are not captured by the quasi-steady aerodynamic model.

6. CONCLUSION

Three aerodynamic models based on blade element momentum theory have been compared to the aerodynamic model of the aero-servo-elastic code HAWC2. The first model assumes quasi-steady aerodynamics. The second model describes the dynamic inflow in a manner which resembles typical engineering models of dynamic inflow and which includes many dynamic states. The third model has been developed for the specific purpose of being suitable for model-based control and state estimation methods. The developed dynamic inflow model, described by only a single dynamic state has proved its ability to model the dynamic inflow in a manner compareable to that of the more advanced models.

It has also been shown that dynamic inflow should be taken into account when estimating the free mean wind speed and the induced velocities. A fact often overlooked or neglected in model-based control and estimation algorithms where the phenomenon is not clearly visible when non-rigid blades, tower, driveshaft as well as tower shadow, wind shear, dynamic stall etc. are included in the simulation. But the phenomenon of dynamic inflow does not dissapear merely because it is not easily identified by visual inspection and as it is a significant physical phenomenon of the wind turbine physics it should be included in the models used for model-based control and state estimation methods.

REFERENCES

1. Øye S. Unsteady wake effects caused by pitch -angle changes. *IEA R&D WECS Joint Action on Aerodynamics of Wind Turbines, 1st Symposium*, London, United Kingdom, 1986; 58–79.
2. Snel H, Schepers J. Engineering models for dynamic inflow phenomena. *Journal of Wind Engineering and Industrial Aerodynamics* 1992; **39**(1-3):267–281.
3. Snel H, Schepers J. Joint investigation of dynamic inflow effects and implementation of an engineering method. *Technical Report ECN-C–94-107*, ECN Wind Energy 1995.
4. van Engelen T, van der Hooft E. Dynamic inflow compensation for pitch controlled wind turbines. *European Wind Energy Conference*, London, United Kingdom, 2004.
5. Hansen MH, Hansen A, Larsen TJ, Øye S, Sørensen P, Fuglsang P. Control design for a pitch-regulated, variable speed wind turbine. *Technical Report Risø-R-1500(EN)*, Risø National Laboratory 2005.
6. Østergaard K, Stoustrup J, Brath P. Linear parameter varying control of wind turbines covering both partial load and full load conditions. *International Journal of Robust and Nonlinear Control* 2009; **19**(1):92–116.
7. Bianchi FD, Mantz RJ, Christiansen CF. Gain scheduling control of variable-speed wind energy conversion systems using quasi-lpv models. *Control Engineering Practice* 2005; **13**(2):247 – 255.
8. Hansen MOL. *Aerodynamics of Wind Turbines*. Second edn., Earthscan, 2008.
9. Sørensen NN, Madsen HA. Modelling of transient wind turbine loads during pitch motion. *European Wind Energy Conference*, Milan, Italy, 2007.
10. Larsen TJ, Hansen AM. How 2 HAWC2, the user's manual. *Technical Report Risø-R-1597(ver. 3-1)(EN)*, Risø National Laboratory 2007.
11. Madsen HA, Mikkelsen R, Øye S, Bak C, Johansen J. A detailed investigation of the blade element momentum (bem) model based on analytical and numerical results and proposal for modifications of the bem model. *Journal of Physics: Conference Series 75: The Science of Making Torque from Wind*, Copenhagen, Denmark, 2007.
12. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW reference wind turbine for offshore system development. *Technical Report NREL/TP-500-38060*, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401-3393 February 2009.
13. Grewal MS, Andrews AP. *Kalman Filtering*. 3rd revised edn., John Wiley and Sons Ltd, 2008.

WIND ENERGY
Wind Energ. 2010; **00**:1–10
DOI: 10.1002/we

RESEARCH ARTICLE

Wind speed and wave force estimation for a floating wind turbine

L. C. Henriksen [1], M. H. Hansen [1], N. K. Poulsen [2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark
[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ABSTRACT

Offshore wind turbines has so far been limited to shallow waters enabling the foundations to be placed on the seabed. Floating wind turbines offer an alternative, as they can be placed at greater water-depths than the conventional offshore wind turbines and are thus more versatile with regards to placement when new wind farms are planned. The added degrees of freedom of a floating structure compared to a fixed structure pose new challenges for model-based control algorithms. In this work a conventional wind turbine model used by model-based control and estimation algorithms is extended to describe the phenomena observed on a floating wind turbine. The work documents the importance of including wave forces in the model used by model-based control and estimation algorithms. Fore-aft tower bending and tower pitch are also shown to be key features that should be included in the model to obtain satisfactory wind speed and wave force estimates. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS

extended kalman filter; hydrodynamic forces; floating horizontal axis wind turbine

Correspondence
L. C. Henriksen, Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark.

E-mail: larh@risoe.dtu.dk

Received . . .

1. INTRODUCTION

Floating wind turbines have become an area increasing interest in the past couple of years. The need to place wind turbines at ever deeper water depths as the existing off-shore sites are filled has pushed the development of novel floating wind turbine concepts. Ref. [1] describes several concepts: a spar buoy, a tension leg platform and a barge platform. The novel concepts has necessitated the need for new control methods adapted to the special challenges posed by floating wind turbines. Among those challenges is the low frequency pitch of the floating structure. Ref. [2] investigated control of the spar buoy concept, originally suggested by Ref. [3] , and concluded that detuning of the blade pitch controller gains gave stable closed-loop performance, as the closed-loop poles was below the tower pitch frequency. Ref. [4] use model-based control of a floating wind turbine based on the barge platform concept, but assume full state information. The assumption of full state information is however not realistic, as the turbulent wind field and waves, resulting in aerodynamic and hydrodynamic forces on the floating wind turbine, cannot easily be measured.

In this work, the basic model for a floating wind turbine is presented and results concerning wind speed and wave force estimations are shown. The presented model utilizes the dynamic inflow model proposed in Ref. [5] and has further extended it to accommodate individual pitch control by estimating the individual wind speeds of each blade. Wind speed estimation on a floating wind turbines is not a novelty as Ref. [6] used an wind speed observer to reduce tower fatigue. The observer presented in this paper differs from that of Ref. [6] by the inclusion of more states, making it suitable for model-based control, and the fact that it is prepared for model-based control with individual pitching of blades.

The paper is organized as follows: The first section describes the model of the floating wind turbine. The next section describes extended Kalman filter theory, which is the underlying method used by the state estimation algorithm. Then

results for different simulations are presented and the ability to estimate the blade specific wind speeds and the wave force are presented and discussed.

2. MODEL

In this section, the different submodels, constituting the floating wind turbine model are introduced. The model contains aerodynamic and hydrodynamic submodels as well as structural models for the tower in the fore-aft direction and drive-shaft. Structural parameters for the tower fore-aft and drive-shaft degrees of freedom have been obtained via a system identifaction method known as prediction error method[7] (PEM). Actuator dynamics have been omitted in this work.

2.1. Aerodynamic model

The aerodynamic model used in this work is based on the single-state dynamic inflow model proposed in Ref. [5] with the extension that the effective wind speed of the individual blades rather than a rotor-wide effective wind speed are included. This extension of the model enables individual pitch control in future work.

For the blades $i = 1, \ldots, n_b$, the aerodynamic torque Q_i and thrust force T_i are the integrated aerodynamic forces over the blade span

$$T_i(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_0^R F_n(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{1}$$

$$Q_i(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_0^R rF_t(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{2}$$

where $F_n(\cdot)$ and $F_t(\cdot)$ are local blade forces normal and tangential to rotor-plane. The local tangential induced velocities are assumed quasi-steady and the local axial induced wind speeds $v_{n,i}(r)$ are given by the averaged induced axial wind speed $\bar{v}_{n,i}$ and the quasi-steady distribution of the axial induction factor

$$v_{n,i}(r) = \frac{a_n^{qs}(\lambda_i, \theta_i, r)}{\bar{a}_n^{qs}(\lambda_i, \theta_i)} \bar{v}_{n,i} \tag{3}$$

where

$$\bar{a}_n^{qs}(\lambda_i, \theta_i) = \frac{1}{R} \int_0^R a_n^{qs}(\lambda_i, \theta_i, r)dr \tag{4}$$

The temporal dynamics of the averaged axial induced velocity is governed by a first order ordinary differential equation

$$\bar{v}_{n,i} = \frac{1}{\tau_i s + 1} V_{rel,i} \bar{a}_n^{qs}(\lambda_i, \theta_i) \tag{5}$$

with the time constant

$$\tau_i = \frac{1}{2} \frac{1.1R}{V_{rel,i} - 1.3\bar{v}_{n,i}} \tag{6}$$

based on the same assumption as in Ref. [5] .

The aerodynamic torque and thrust of each blade comprises the rotor-wide entities

$$Q = \sum_i^{n_b} Q_i \text{ and } T = \sum_i^{n_b} T_i \tag{7}$$

Strain gauges (SG) are placed at root of the blade at the radial distance r_{SG} and projected into the normal and tangential directions of the rotor plane. If the blades where assumed rigid and massless the aerodynamic bending moments

$$Q_{n,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_{r_{SG}}^R rF_n(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{8}$$

$$Q_{t,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_{r_{SG}}^R rF_t(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{9}$$

Wind Energ. 2010; **00**:1–10 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

could directly be obtained from the strain gauges and used to estimate blade specific wind speeds. However, the measured out-of-plane root bending moment $Q_{n,i}^{SG}$ is affected by the centrifugal loading of the blades and is thus lower than the solely aerodynamic out-of-plane root bending moment. Several methods exist to make the out-of-plane sensor usable in the estimation algorithm: The model can be extended to include the structural properties of the blade or the estimation algorithm can assign smaller importance of the sensor by increasing the expected output variance of the sensor used to tune the estimation algorithm. The measured in-rotor-plane root bending moment $Q_{t,i}^{SG}$ is heavily influenced by the gravitational loading of the blades varying with the azimuth angle of the blades. The aerodynamic moments are approximated by the blade root bending moments in the following way

$$Q_{n,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) \approx Q_{n,i}^{SG} + \nu_{qn,i}, \qquad \nu_{qn,i} \in N(0, \sigma_{qn}^2) \tag{10}$$

$$Q_{t,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) \approx Q_{t,i}^{SG} - F_{g,blade} \sin\phi_{b,i} + \nu_{qt,i}, \qquad \nu_{tn,i} \in N(0, \sigma_{qt}^2) \tag{11}$$

where $F_{g,blade}$ is the gravitational loading on the individual blades and $\phi_{b,i}$ is the azimuth angle of the individual blades and the zero-mean Gaussian distributed noise contributions $\nu_{qn,i}$ and $\nu_{qt,i}$ represent the uncertainties caused by blade dynamics, blade deformation, centrifugal stiffening etc.

2.2. Wind spectra

For rotor-wide wind speed spatially averaged models, based on e.g. the turbulence model of Ref. [8] , are suitable. But for the individual wind speeds of each blade, the strongest source of wind speed variation is the periodic contribution caused by wind shear, tower shadow and the spatial structure of the turbulent field [9]. As a result the wind variation is modeled as a periodically varying wind speed depending on the rotation speed of rotor. The total wind speed for blade i is the sum of a rotor-wide mean wind speed V_m and a blade specific turbulent wind speed $V_{t,i}$

$$V_i = V_m + V_{t,i} \tag{12}$$

where the turbulent wind speed spectrum is given by the rotational speed of the rotor

$$V_{t,i} = \frac{\omega_{n,v}^2}{s^2 + 2\omega_{n,v}\zeta_{v,i}s + \omega_{n,v}^2}\xi_{v,i}, \quad \xi_{v,i} \in N(0, \sigma_v^2) \tag{13}$$

The damped frequency of the turbulent wind field is equal to the rotational speed of the rotor $\omega_{d,v} = \Omega$. Giving a natural frequency dependent of damped frequency and damping ratio $\omega_{n,v} = \omega_{d,v}/\sqrt{1-\zeta_v^2}$. For a constant periodic wind speed the damping ratio should ideally be zero, numerical conditioning especially with regards to the synthesis of model-based controllers, e.g. LQ, is however eased by adding a little damping in the model and $\zeta_v = 0.01$ is chosen. The rotor-wide mean wind speed is modeled as a parameter rather than a state and a first order filter driven by the mean of the individual wind speed estimates of the current sample determines the mean wind speed in the next sample

$$V_m = \frac{1}{\tau s + 1}V, \quad V = \frac{1}{n_b}\sum_i^{n_b} V_i \tag{14}$$

where a time constant $\tau = 1/(\Omega \text{ rad/s})$ has shown good performance with regards to the estimation of the blade specific wind speeds.

2.3. Hydrodynamic forces

The offshore tower either in a floating or fixed configuration is subjected to wave forces. If these forces are not taken into account, the tower top displacement will be ascribed solely to the aerodynamic thrust resulting in a erroneous estimation of wind speed. The hydrodynamic forces on a moving body with velocity v in a fluid with the oscillating flow velocity u can be described by the semi-empirical Morrison equation [1] .

$$F_h = \rho_w V_b \dot{u} + \rho(C_m - 1)V_b(\dot{u} - \dot{v}) + \frac{1}{2}\rho_w C_d A_b(u - v)|u - v| \tag{15}$$

where C_m and C_d are the inertia and drag coefficients of the floating body, ρ_w is the mass density of the fluid and V_b and A_b are volume and area of the body submerged in the fluid. The Morrison equation can be simplified to

$$F_h = \kappa_1\dot{u} + \kappa_2\dot{v} + \kappa_3(u - v)|u - v| \tag{16}$$

where the constants κ_1,κ_2 and κ_3 contain the aforementioned coefficients, volume etc. The third term describing the drag force, is assumed to be negligible, leading to $\kappa_3 = 0$. The first term is rewritten to a wave force $F_w = \kappa_1\dot{u}$. The modeled

spectrum of the stochastic oscillating wave force is fitted to the measured spectrum of the water acceleration

$$F_w = \frac{k\omega_{n,w}^2}{s^2 + 2\omega_{n,w}\zeta_w s + \omega_{n,w}^2}\xi_w, \quad \xi_w \in N(0, \sigma_w^2) \tag{17}$$

giving a damped frequency $\omega_{d,w} = \frac{2\pi}{T_p}$ defined by the peak spectral period T_p. In the work of Ref. [10] values for the peak spectral period and significant wave height are given as $T_p = 10$ s and $H_s = 6$ m and crude fitting of the water acceleration power spectral density function to the model (17) gives a damping ratio $\zeta_w = 0.125$ and a steady state gain k of 0.2. The natural frequency can then be determined as $\omega_{n,w} = \omega_{d,w}/\sqrt{1-\zeta_w^2}$. The stochastic input w_w is assumed to be zero-mean Gaussian distributed white noise. Simulations (not presented here) have shown that the wave force estimation is not sensitive to whether or not the correct peak spectral period is known, correct information about the peak spectral period is of higher importance when used in a model-based control algorithm. The correct peak spectral period can be estimated online via the estimated wave force. The peak of the power spectral density can by computed using various algorithms e.g. Fast Fourier Transform (FFT) algorithms. The peak spectral period can then be updated, e.g. through a first order dynamic filter with a time constant in the range of tens of minutes.

2.4. Tower and spar buoy

The fore-aft motion of the wind turbine can be comprised of three degrees of freedom: Tower bending and pitch and surge of the entire wind turbine. Pitch describes the angle which the turbine is rotated with regards to the vertical axis and surge indicates the displacement of the wind turbine in the wind direction, see also Fig. 1 for a depiction of the mentioned degrees of freedom. In this work, the wind turbine pitch (*tp*) angle is defined as the pitch angle of the wind turbine at the bottom, denoted α. For a rigid tower and spar buoy, the wind turbine pitch angle is related to the relative displacement of the top and bottom of the wind turbine by a simple trigonometric relation

$$y_{tp} = h \sin\alpha \tag{18}$$

where h is the height of the wind turbine. For small variations of α, the pitch displacement be approximated as

$$y_{tp} \approx h\alpha \tag{19}$$

In this work, the surge displacement y_{surge} is defined as the displacement of the bottom of the floating wind turbine relative to point where the turbine would be if no wind or waves where affecting it. The relative displacement of the tower top, where turbine pitch y_{tp} and tower bending (*tb*) y_{tb} are included, is denoted y_{rel} and is given as

$$y_{rel} = y_{top} - y_{surge} = y_{tp} + y_{tb} \tag{20}$$

The effects of bending, pitch and surge can be observed in Fig. 2, which depicts the results of a simulation. Simulation details can be found in the Results section of this paper. In the first plot it is seen that the difference between y_{rel} and y_{tp} is small compared to the absolute variations of y_{rel} and y_{tp}. The second plot shows that the pitch degree of freedom given by $\dot{y}_{rel}$ and $\dot{y}_{tp}$ is approximately within a ± 2 m/s operating range. The surge degree of freedom, which is the difference between $\dot{y}_{top}$ and $\dot{y}_{rel}$, has an approximately 10 times smaller contribution to the fore-aft velocity of the wind turbine compared to the pitch degree of freedom and is omitted from the model as it is considered negligible. The third plot shows a significant opposite phase between top and bottom measurements and is caused by bending of the tower, as the tower is not rigid but flexible.

The tower bending and pitch of the floating wind turbine can be described by two interconnected mass-damper-spring systems (M_{tp}, D_{tp}, K_{tp}) and (M_{tb}, D_{tb}, K_{tb}) where the first system models the pitch motion of the wind turbine and the second system models the bending of the tower. The top of the wind turbine is affected by the aerodynamic thrust forces of each blade T_i, which are functions of the relative wind speed seen from the wind turbine top $V_{rel,i} = V_i - \dot{y}_{rel}$. The bottom of the wind turbine is affected by the hydrodynamic forces (16) consisting of a wave force and a turbine acceleration term. A turbine acceleration term $\kappa_2\ddot{y}_\alpha$ can be added to the inertial mass of the turbine pitch resulting in a new mass $\tilde{M}_{tp} = M_{tp} + \kappa_2$ and the resulting system

$$M_{tb}\ddot{y}_{rel} + D_{tb}\dot{y}_{tb} + K_{tb}y_{tb} = T \tag{21a}$$

$$\tilde{M}_{tp}\ddot{y}_{tp} + D_{tp}\dot{y}_{tp} + K_{tp}y_{tp} = F_w + D_{tb}\dot{y}_{tb} + K_{tb}y_{tb} \tag{21b}$$

describes the fore-aft degrees of freedom of the floating wind turbine, see Fig. 1 for a depiction of the two interconnected mass-damper-spring systems. It is expected that an acceleration sensor in the tower top and an angle measurement in the bottom are realistic assumptions of available sensors. The velocity and acceleration of the bottom angles can then be calculated via numerical difference.

Wind Energ. 2010; **00**:1–10 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

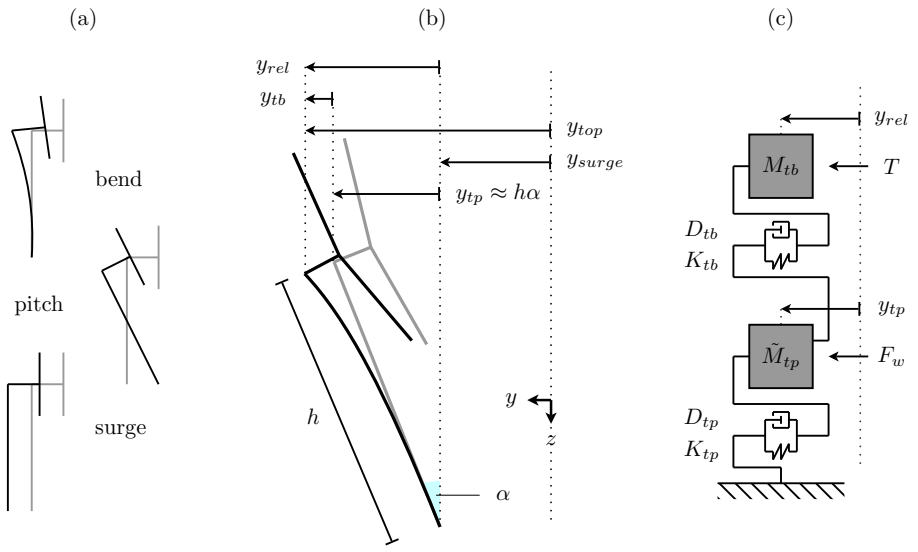


Figure 1. Degrees of freedom of tower in fore-aft direction. (a) Degrees of freedom: tower bend, pitch and surge, only tower bend and pitch are included in the model. (b) The relative position y_{rel} is the difference between the position of the tower top y_{top} and the floater bottom y_{surge}. The sum of turbine pitch displacement y_{tp} and tower bending displacement y_{tb} equals y_{rel}. (c) The floating tower modeled as two interconnected mass-damper-spring systems.

2.5. Drive-train

The drive-train connects the rotor to the generator through a low speed shaft, a gearbox and a high speed shaft. The drive-train flexibility is a modeled in the low speed shaft coordinate system

$$I_r\ddot{\phi}_r + D_s\dot{\phi}_\Delta + K_s\phi_\Delta = Q \tag{22}$$

$$I_g N_g^2 \frac{\ddot{\phi}_g}{N_g} - D_s\dot{\phi}_\Delta - K_s\phi_\Delta = -Q_g N_g \tag{23}$$

where N_g is the gear ratio, I_r and I_g are the moments of inertia of the rotor and generator, K_s and D_s are the spring and damping constants. It should also be mentioned that the following definitions are introduced: $\dot{\phi}_r \equiv \Omega$ is rotor speed, $\dot{\phi}_g \equiv \Omega_g$ is generator speed and $\phi_\Delta \equiv \phi_r - \frac{\phi_g}{N_g}$ is the angular torsion of the drive-shaft in the low speed shaft coordinate system.

3. EXTENDED KALMAN FILTER

The ordinary differential equations of the previous section can be gathered in an ordinary differential state space equation and an output equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t) \tag{24a}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{v}(t) \tag{24b}$$

The ordinary differential state space equation can be time-discretized giving the state progress equation

$$\boldsymbol{x}(t_{k+1}) = \underline{\boldsymbol{f}}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)) + \boldsymbol{w}(t_k) \tag{25}$$

States, outputs and inputs are

$$\boldsymbol{x} = [\Omega\ \Omega_g\ \phi_\Delta\ y_{tp}\ \dot{y}_{tp}\ y_{tb}\ \dot{y}_{tb}\ V_{t,i}\ \dot{V}_{t,i}\ \bar{v}_{n,i}\ F_w\ \dot{F}_w]^T \tag{26}$$

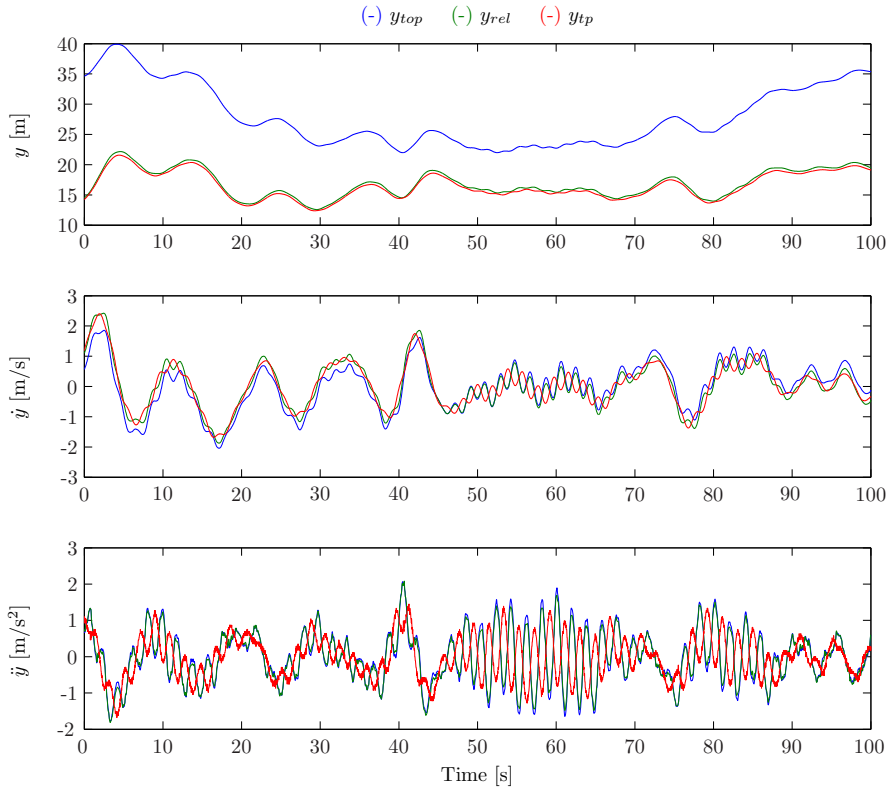


Figure 2. Floating tower fore-aft motion - The first plot shows tower top position shown with 3 different sensors: 1. absolute position in y direction, 2. position relative to position of bottom of floater in y direction, 3. absolute pitch angle multiplied with height of wind turbine. The second plot depicts the velocities of the 3 sensors. It can be seen that low frequency displacement of the entire structure (surge) has little impact on the velocity measurements. The third plot display the accelerations and it can be seen that the tower bending mode causes the top and bottom sensors to differ, as they are in opposite phase.

$$\boldsymbol{y} = [\Omega\ \Omega_g\ \phi_\Delta\ \alpha\ \dot{\alpha}\ \ddot{\alpha}\ \ddot{y}_{top}\ P_e\ Q_{n,i}^{SG}\ Q_{t,i}^{SG}]^T \tag{27}$$

$$\boldsymbol{u} = [\theta_i\ Q_g]^T \tag{28}$$

where $i = 1, \ldots, n_b$. The process and measurement noise vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ are zero-mean Gaussian distributed with the covariance matrices $\mathbf{R_x}$ and $\mathbf{R_y}$, respectively. The process and measurement noise vectors are given as

$$\boldsymbol{w} = \boldsymbol{w_0} + \mathbf{G_w}\boldsymbol{w_1} \text{ and } \boldsymbol{v} = \boldsymbol{v_0} + \mathbf{G_v}\boldsymbol{v_1} \tag{29}$$

where $\boldsymbol{w_0}$ and $\boldsymbol{v_0}$ are background low-level zero-mean Gaussian distributed noise with the covariance matrices $\mathbf{R_{x,0}}$ and $\mathbf{R_{y,0}}$, respectively. The additional process and measurement noise vectors $\boldsymbol{w_1} = [\xi_{v,i}\ \xi_w]^T$ and $\boldsymbol{v_1} = [\nu_{qn,i}\ \nu_{qt,i}]^T$ are added to the general noise vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ through the noise gain matrices $\mathbf{G_w}$ and $\mathbf{G_v}$, respectively. The general noise covariance matrices are given as

$$\mathbf{R_x} = \mathbf{R_{x,0}} + \mathbf{G_w}\text{diag}([\sigma_v^2\ \ldots\ \sigma_v^2\ \sigma_w^2])\mathbf{G_w}^T \tag{30}$$

$$\mathbf{R_v} = \mathbf{R_{v,0}} + \mathbf{G_v}\text{diag}([\sigma_{qn}^2\ \ldots\ \sigma_{qn}^2\ \sigma_{qt}^2\ \ldots\ \sigma_{qt}^2])\mathbf{G_v}^T \tag{31}$$

An extended Kalman filter [11] has been employed to estimate the states of the model presented in the previous section, among those states are the effective wind speeds at each blade and the wave force state. The estimated states are denoted

$\hat{\boldsymbol{x}}_{k_a|k_b}$, where $k_a|k_b$ means the estimation at time k_a given by the information available at time k_b. The *a-posteriori* estimation of the states is given by

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{L}_k[\boldsymbol{y}_k - \boldsymbol{g}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k)] \tag{32}$$

Enabling an *a-priori* estimation of the one-step-ahead prediction of the states

$$\hat{\boldsymbol{x}}_{k+1|k} = \underline{\boldsymbol{f}}(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k) \tag{33}$$

where the Kalman gain $\mathbf{L}_k$ and output error covariance $\mathbf{\Psi}_k$

$$\mathbf{L}_k = \mathbf{P}_{k|k-1}\mathbf{C}_{k|k-1}^T\mathbf{\Psi}_k^{-1} \tag{34a}$$

$$\mathbf{\Psi}_k = \mathbf{C}_{k|k-1}\mathbf{P}_{k|k-1}\mathbf{C}_{k|k-1}^T + \mathbf{R}_\mathbf{y} \tag{34b}$$

are updated by the discrete time recursive Riccati equation

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k\mathbf{C}_{k|k-1}\mathbf{P}_{k|k-1} \tag{34c}$$

$$\mathbf{P}_{k+1|k} = \underline{\mathbf{A}}_{k|k}\mathbf{P}_{k|k}\underline{\mathbf{A}}_{k|k}^T + \mathbf{R}_\mathbf{x} \tag{34d}$$

The EKF is based on an extension of the Kalman filter, which is the optimal estimator for linear systems. Consequently the time discrete model consisting of (25) and (24b) is linearized w.r.t. $\boldsymbol{x}$ using first order Taylor series approximation giving $\underline{\mathbf{A}}_{k|k}$, which is the Jacobian of the state progress equation $\underline{\boldsymbol{f}}(\cdot)$ linearized at $(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k)$ and $\mathbf{C}_{k|k-1}$, which is the Jacobian of the output equation $\boldsymbol{g}(\cdot)$ linearized at $(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k)$.

4. RESULTS

A simulation is performed with a turbulent wind field [8] with turbulence intensity 0.14, power law wind shear with a exponent of 0.14 and potential flow tower shadow. Linear irregular wave kinematics based on composition of Airy waves with Wheeler stretching, with a significant wave height H_s = 6 m and peak spectral period T_p = 10 s, are used. The chosen parameters for the hydrodynamic loads are based on Ref. [10] . A collective blade pitch gain scheduled PI controller similar to the one of Ref. [1] has been used to control the floating wind turbine. The simulation is performed with the hydro-aero-servo-elastic simulations software HAWC2[12] developed by Risø DTU.

The wind speed and wave force estimation performance of the extended Kalman filter based on the model presented in this paper can be seen in Fig. 3. The first subplot in Fig. 3 depicts the wind speed V, which is the average of the blade specific rotationally sampled wind speeds as the blades pass through tower shadow, wind shear and the turbulent features of the wind field. It can be seen that the real wind speed and the estimated wind speed are in good agreement, except for an offset largely caused by blade deformations, blade cone and rotor tilt, all of which are not included in the BEM model used by the extended Kalman filter. The estimated wind speed lags behind the real wind speed with a few seconds, the estimation lag is caused by the fact that the Kalman filter can not be tuned to hard due to model uncertainties etc. The better that model and plant are in agreement, the harder the Kalman filter can be tuned and the estimation lag would be reduced as a result. The second subplot shows the relative wind speed V_{rel} given as $V - \dot{y}_\beta$, the wind speed and its estimated counter part are in good agreement apart from the phase lag and offset. A scaled estimation of the wave force F_w is compared to the water acceleration $\dot{u}$ at 10 m depth below the mean sea level in the third subplot. The water acceleration and estimated wave force a in general in good agreement. From approximately 50 s to 65 s a significant high frequency contribution on the wave force estimation is observed, this coincides with significant tower bending as observed in Fig. 2, indicating that the aerodynamic thrust and wave forces are not fully separable by the extended Kalman filter in its current configuration. The inability to separate the aerodynamic thrust and wave forces are likely due to model uncertainties e.g. unmodeled phenomena such as blade deformations which are exited by the significant tower bending occuring at the same time.

In Fig. 4 the wind speed signal has been binned with regards to the azimuth angle of the blades and it can be seen once more that the estimations have a slight offset and phase lag compared to the real signals. The phase lag of the estimated wind is approximately 35 degrees and individual pitch control would probably not achieve good results in terms of load reduction with a phase lag of that size. Preliminary investigations, not presented in this paper, suggest that the wind speed estimation performance could be improved by the inclusion of blade structural degrees of freedom in the model used by the Kalman filter. The improvement would increase chances of a successful implementation of individual pitch control in future work.

Depending of which sensors are used and depending on how much the particular sensors are trusted, different wind speed estimation performance is observed in the simulations, the same applies for different levels of modeling detail. Fig.

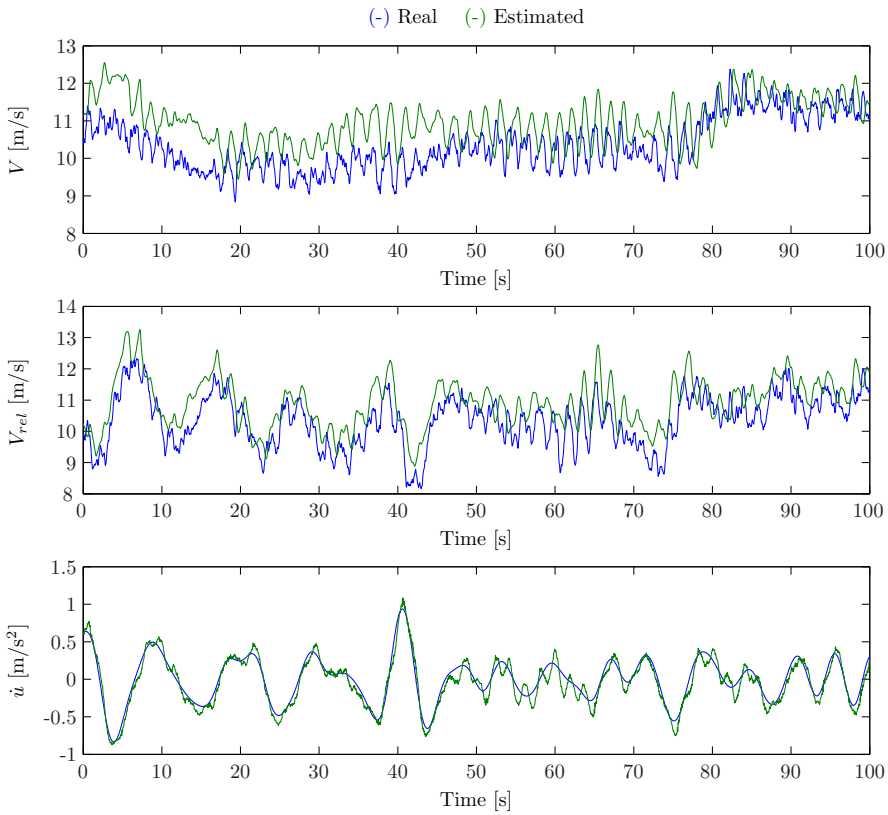


Figure 3. The first subplot depicts the wind speed observed by the rotating blades. The second subplot shows the relative wind speed. The third subplot displays the scaled wave force estimates and the acceleration of the water at 10 m depth.

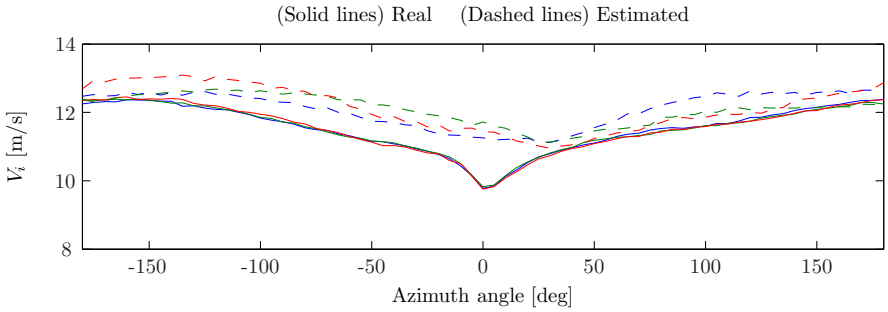


Figure 4. Individual wind speeds for each blade and estimations thereof binned with regards to the azimuth angle of the individual blades.

5 illustrates the importance of including significant phenomena in the model used by the extended Kalman filter. Three different models are used by the extended Kalman filter: The first model is the same model as used in Fig. 3 including tower bending and wave forces. The second model differs from the first by excluding the wave force contribution in the model. The third model differs from the first by omitting tower bending in the model. The wave forces are clearly seen to influence the wind speed estimations of the extended Kalman filter based on the second model and the tower bending is

Wind Energ. 2010; **00**:1–10 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

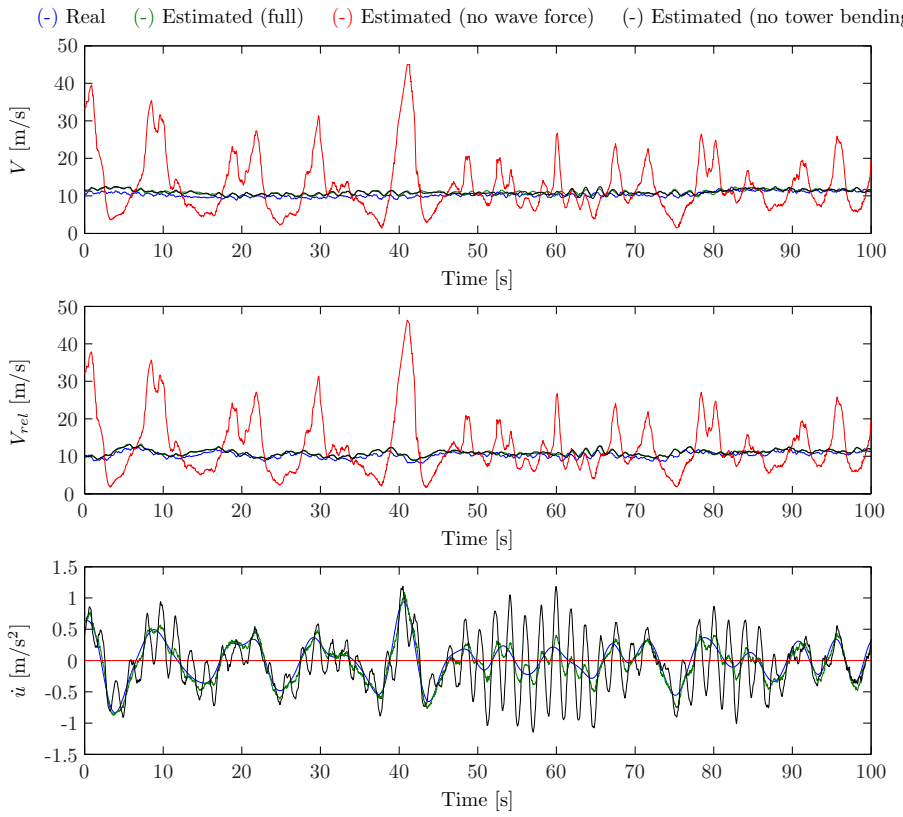


Figure 5. Wind speed and wave force estimation by extended Kalman filters based on three different models: The first model includes all the submodels presented in this work, including tower bending and wave force contribution. The second model omits wave forces and the third model excludes tower bending.

clearly seen to disturb the wave force estimation of the extended Kalman filter based on the third model. All three Kalman filters are tuned with similar covariance matrices and the tuning is quite ”aggressive” making the estimations sensitive to modeling errors. A detuning of the Kalman filters rendering them less aggressive would somewhat dampen the obvious poor performance of the Kalman filters based on the second and third models, but the problem remains: The second and third model does not the describe the floating wind turbine adequately.

5. CONCLUSION

In the present work the significant phenomena of a floating wind turbine in the fore-aft direction have been identified and an extended Kalman filter based on the presented model has been used to estimate wind speed and wave forces. Wave force contribution as well as tower pitch and tower bending has been identified as important submodels, which should all be included in the model in order to achieve good wind speed and wave force estimation performance. Furthermore, the presented model enables blade specific wind speed estimates, which in turn enable individual pitch control when used in a model-based control algorithm. Blade deformations has not been included in the presented model and it remains to be seen if further improvements regarding wind speed and wave force estimations could be obtained by extending the model to include blade structural degrees of freedom.

REFERENCES

1. Jonkman J. Dynamics modeling and loads analysis of an offshore floating wind turbine. *Technical Report NREL/TP-500-41958*, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401-3393 November 2007.
2. Larsen TJ, Hanson TD. A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine. *J. Phys.: Conf. Ser.* 2007; **75 012073 (11pp)**:doi:10.1088/1742–6596/75/1/012 073.
3. Nielsen FG, Hanson TD, Skaare B. Integrated dynamic analysis of floating wind turbines. *Proceedings of OMAE2006*, 2006.
4. Namik H, Stol K. Individual blade pitch control of floating offshore wind turbines. *Wind Energy* 2010; **13**(1):74–85.
5. Henriksen LC, Hansen MH, Poulsen NK. The effect of dynamic inflow in free mean wind speed estimation. *Submitted for Wind Energy* 2010; .
6. Skaare B, Hanson TD, Nielsen FG. Importance of control strategies on fatigue life of floating wind turbines. *ASME Conference Proceedings* 2007; **2007**(42711):493–500.
7. Ljung L. *System Identification: Theory for the User*. 2nd edn., Prentice Hall, 1999.
8. Mann J. Wind field simulation. *Probabilistic Engineering Mechanics* 1998; **13**(4):269–282.
9. Kristensen L, Frandsen S. Model for power spectra of the blade of a wind turbine measured from the moving frame of reference. *Journal of Wind Engineering and Industrial Aerodynamics* 1982; **10**(2):249 – 262, doi:DOI: 10.1016/0167-6105(82)90067-8.
10. Passon P, Kuhn M, Butterfield S, Jonkman J, Camp T, Larsen T. Oc3-benchmark exercise of aero-elastic offshore wind turbine codes. *J. Phys., Conf. Ser. (UK)* 2007; **75**(1):–.
11. Grewal MS, Andrews AP. *Kalman Filtering*. 3rd revised edn., John Wiley and Sons Ltd, 2008.
12. Larsen TJ, Hansen AM. How 2 HAWC2, the user's manual. *Technical Report Risø-R-1597(ver. 3-1)(EN)*, Risø National Laboratory 2007.

Part II

Implemented Controllers

WIND ENERGY
Wind Energ. 2010; **00**:1–16
DOI: 10.1002/we

RESEARCH ARTICLE

Wind Turbine Control with Constraint Handling: A Model Predictive Control approach

L. C. Henriksen [1], M. H. Hansen [1], N. K. Poulsen [2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark
[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ABSTRACT

This paper presents a wind turbine controller able to handle both hard and soft constraints, typically on actuators but also on other components of the wind turbine, if needed. An issue especially relevant during extreme events or for under-dimensioned actuators. The presented controller is based on model predictive control, a control method well suited for constraint handling. The performance of the presented controller during an extreme operating gust is compared to that of a proportional-integral controller with integrator anti-windup. Furthermore, the presented controller's capability to operate in the full wind speed range is demonstrated by additional test cases. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS

model predictive control; constraint handling; horizontal axis wind turbine

Correspondence
L. C. Henriksen, Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark.

E-mail: larh@risoe.dtu.dk

Received . . .

1. INTRODUCTION

In the pursuit of lowering the cost-to-power ratio of wind turbines, advanced control algorithms play an important role. Model based control algorithms such as LQG, H_∞, LPV, nonlinear controllers etc. and their application to wind turbines have all been covered thoroughly e.g. [1, 2, 3]. These methods are attractive because the control objectives can be prioritized such as power production and load reduction of key components. Via observers, the unmeasured states can be estimated and the integration of new types of sensors in conjunction with well known sensors can be handled reasonably easy within the model based framework.

The algorithms mentioned above have however usually one shortcoming: The ability to handle constraints in a systematic way, the solution is typically to make the controller robust and thus decrease the potential performance of the controller. Constraints can be either physical limitations of e.g. the actuators, such as limits of the pitch angle [4] or artificial constraints, such as the rate of change of the pitch angle which should be within some limits to prevent excessive fatigue damage [5]. Other types of artificial constraints, such as simplified fatigue models of selected key components [6, 5], could be included in the control design model and effort could be put to keeping the approximated fatigue expressions within some predefined limits.

Model predictive control (MPC) offers the ability to handle constraints. The application of MPC on wind turbines have previously been investigated in [7, 5]. The ability to handle the nonlinear physics of wind turbines is investigated in [7] but no special attention is given to constraints. Damage mitigation control is investigated in [5] however only operation above rated wind speed is considered.

The present work uses MPC to respect the constraints on the actuators, i.e. pitch angle and -rate and generator torque and -rate, which in this case are considered to be real and hard constraints. It is beyond the scope of this paper to investigate the fatigue of key mechanical structures such as drive-shaft torsion or tower bending. The primary focus is on constraint handling and effort will thus not be put into comparing fatigue loads with different controllers but simply verify the ability

Prepared using weauth.cls [Version: 2010/06/17 v1.00]

to control a wind turbine subjected to constraints, both physical and artificial. To give a simplified example, artificial constraints on the generator speed and -power are added to the control algorithm and investigated during an extreme operating gust. The artificial constraints will be at selected percentages above the rated values of the generator power and speed. During extreme events, the violation of these artificial constraints can be prioritized such that the violation of one constraint is allowed if it decreases the violation of another more important constraint. Although the given example might not apply for real wind turbines, the presented principle can be applied to more useful objectives such as damage mitigation control [5].

In this paper, MPC within a hybrid controller framework is presented. The hybrid controller framework ensures switching between the different regions of operation and enables operation over the entire wind speed range. State and disturbance observers are used by the MPC to estimate the states and the unmeasurable disturbances. The inherent use of observers to estimate the disturbances also handles the problem of not being able to measure all the states used by the MPC.

The wind turbine in this paper is inspired by the National Renewable Energy Laboratory (NREL) 5MW reference wind turbine [8], which is a variable speed and pitch controlled wind turbine. The MPC is compared to a PI controller, also presented in [8]. The PI in this paper has however been to modified to be able to handle saturations of the pitch rate such that a fair comparison between the MPC and PI can be performed. The control methods presented in this paper are applicable for the class of wind turbines equipped with variable speed generators and controlled by the pitch of the rotor blades, the electromagnetic torque of the generator and the yaw drive of the nacelle. The latter is however not addressed in this work, as perfect yaw alignment is presumed.

This paper is organized in the following manner: The first section introduces the governing equations of the wind turbine model, necessary for the synthesis of model-based control algorithms. The second section presents the concept of the hybrid controller setup which covers the handling of the different regions of operation depending on the mean wind speed. The third section describes model predictive control which is the underlying methodology in the individual controllers of the hybrid controller. Finally, results of simulation are shown and discussed.

2. MODEL FOR CONTROLLER DESIGN

The control design model is given by a number of algebraic and differential equations. The primary algebraic equation is the aerodynamic power transferred from wind to rotor

$$P_r = \tfrac{1}{2}\rho\pi R^2 V^3 C_P(\theta,\lambda) \tag{1}$$

where ρ is the mass density of air, R is the rotor disc radius, V is the relative effective mean speed over the entire rotor disc. The aerodynamic power coefficient C_P determining how much of the total amount of available power in the rotor disc is transferred from the wind to the rotor. C_P is a function of the blade pitch angle θ and the tip-speed-ratio λ, which is the ratio between the speed of the blade at the tip and the wind speed

$$\lambda = \frac{\Omega_r R}{V} \tag{2}$$

The wake is assumed to be in equilibrium, i.e., the dynamic inflow effects of the induced velocities from the wake are neglected. The aerodynamic power P_r relates to the aerodynamic torque Q_r and rotor speed Ω_r

$$P_r = \Omega_r Q_r \tag{3}$$

The aerodynamic thrust force Q_t induced by the mass flow of air around rotor blades and tower in the wind direction is given by

$$Q_t = \tfrac{1}{2}\rho\pi R^2 V^2 C_T(\theta,\lambda) \tag{4}$$

where the aerodynamic thrust coefficient C_T is a function of the blade pitch and tip-speed-ratio. The mechanical power P_g transferred from the high speed drive-shaft to the generator relates to the generator speed Ω_g and the generator torque Q_g

$$P_g = \Omega_g Q_g \tag{5}$$

All losses in drive-shaft, gearbox, bearings, generator etc. are lumped into a parameter and transferred into the conversion from mechanical power to electrical power P_e in the generator with a efficiency η

$$P_e = \eta P_g \tag{6}$$

Wind Energ. 2010; **00**:1–16 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

The relative wind speed average over the rotor disc due to the velocity of nacelle displacement $\dot{\psi}_t$ is given by

$$V = V_{eff} - \dot{\psi}_t \tag{7}$$

where the effective wind speed V_{eff} is the sum of the mean wind speed V_m and a spatial averaged stochastic turbulent wind speed V_t

$$V_{eff} = V_m + V_t \tag{8}$$

To have a simple turbulence model, for the rotor disc, the spectrum of a spatial averaged Mann turbulence [9, 10] is fitted to a second order linear filter, with the time constants τ_1 and τ_2, for different mean wind speeds

$$V_t = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e, \quad e \in N(0, 1) \tag{9}$$

where e is zero-mean Gaussian distributed white noise. The steady state gain of the linear turbulence model is given by $k = \sigma_k \sqrt{2L_k/V_m}$ where L_k is the roughness length and σ_k is the standard deviation, in the main wind direction, of the Kaimal turbulence model, both given in IEC 61400-1 [11].

The following differential equations constitutes the structural part of the design model. The first component is a flexible slow speed drive-shaft and an assumed rigid high speed drive-shaft

$$I_r \ddot{\phi}_r + D_s \dot{\phi} + K_s \phi = Q_r \tag{10}$$

$$I_g N_g^2 \frac{\ddot{\phi}_g}{N_g} - D_s \dot{\phi} - K_s \phi = -Q_g N_g \tag{11}$$

where N_g is the gear ratio, I_r and I_g are the moments of inertia of the rotor and generator, respectively. K_s and D_s are the spring and damping constants of the low speed drive-shaft. It should also be mentioned that the following definitions are introduced: $\dot{\phi}_r \equiv \Omega_r$ is rotor speed, $\dot{\phi}_g \equiv \Omega_g$ is generator speed and $\phi \equiv \phi_r - \frac{\phi_g}{N_g}$ is the slow speed shaft angular torsion. Another significant model component is the fore-aft motion of the tower

$$M_t \ddot{\psi}_t + D_t \dot{\psi}_t + K_t \psi_t = Q_t \tag{12}$$

where ψ_t denotes the nacelle displacement in the fore-aft direction and is positive in the wind direction, M_t denotes the tower, rotor and nacelle equivalent mass and K_t and D_t denotes the tower spring and damping constants.

The actuators are assumed linear under the assumption that a low level controller, e.g. PID or some type of nonlinear controller e.g. [12], is operating in closed loop with the actuator mechanics. The pitch actuator dynamics, including local low level controller, are described by a second order ordinary differential equation, an approximation which under the proper conditions can be justified [13]

$$\ddot{\theta} + 2\zeta_\theta \omega_\theta \dot{\theta} + \omega_\theta^2 \theta = \omega_\theta^2 \theta_{ref} \tag{13a}$$

subject to

$$\begin{bmatrix} \theta_{min} \\ \dot{\theta}_{min} \end{bmatrix} \leq \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \leq \begin{bmatrix} \theta_{max} \\ \dot{\theta}_{max} \end{bmatrix} \tag{13b}$$

where ω_θ and ζ_θ are the natural frequency and damping ratio of the actuator and θ_{ref} is the reference signal from the controller. The generator torque actuator is described by a first order ordinary differential equation [14]

$$\dot{Q}_g + \tau_g^{-1} Q_g = \tau_g^{-1} Q_{g_{ref}} \tag{14a}$$

subject to

$$\begin{bmatrix} Q_{g,min} \\ \dot{Q}_{g,min} \end{bmatrix} \leq \begin{bmatrix} Q_g \\ \dot{Q}_g \end{bmatrix} \leq \begin{bmatrix} Q_{g,max} \\ \dot{Q}_{g,max} \end{bmatrix} \tag{14b}$$

where τ_g is the time constant of the generator torque actuator and $Q_{g_{ref}}$ is the reference signal from the controller.

The components can be gathered into a nonlinear model composed of a nonlinear ordinary differential equation $\boldsymbol{f} : \mathbb{R}^{n+n_u} \to \mathbb{R}^n$, a state noise matrix $\mathbf{G} \in \mathbb{R}^{n \times n_e}$ and a measurement equation $\boldsymbol{g} : \mathbb{R}^{n+n_u} \to \mathbb{R}^{n_y}$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \mathbf{G}\boldsymbol{w}(t) \tag{15a}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{v}(t) \tag{15b}$$

that describes the relationship between the state vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$, the input vector $\boldsymbol{u} \in \mathbb{R}^{n_u}$, the output vector $\boldsymbol{y} \in \mathbb{R}^{n_y}$ and the state and measurement noise vectors $\boldsymbol{w} \in \mathbb{R}^{n_w}$ and $\boldsymbol{v} \in \mathbb{R}^{n_y}$, respectively, of the complete design model. These vectors contain the following variables

$$\begin{aligned}
\boldsymbol{x} &= [\Omega_r \; \Omega_g \; \phi \; \psi_t \; \dot{\psi}_t \; V_{eff} \; \dot{V}_{eff} \; \theta \; \dot{\theta} \; Q_g]^T \\
\boldsymbol{u} &= [\theta_{ref} \; Q_{g,ref}]^T \\
\boldsymbol{w} &= [e]^T \\
\boldsymbol{y} &= [\Omega_r \; \Omega_g \; \ddot{\psi}_t \; \theta \; \dot{\theta} \; \ddot{\theta} \; Q_g \; \dot{Q}_g \; P_e]^T
\end{aligned}$$

The nonlinear model can be linearized using first order Taylor series approximation where δ denotes small variations away from the linearization point, which in this case is equivalent with equilibrium points for a given wind speed. The linearized model is time-discretized by the zero-order-hold method [15] into the form

$$\delta \boldsymbol{x}_{k+1} = \underline{\mathbf{A}} \delta \boldsymbol{x}_k + \underline{\mathbf{B}} \delta \boldsymbol{u}_k + \underline{\mathbf{G}} \boldsymbol{w}_k, \quad \boldsymbol{w} \in N(\mathbf{0}, \mathbf{R}_\mathbf{w}) \tag{16a}$$

$$\delta \boldsymbol{y}_k = \mathbf{C} \delta \boldsymbol{x}_k + \mathbf{D} \delta \boldsymbol{u}_k + \boldsymbol{v}_k, \quad \boldsymbol{v} \in N(\mathbf{0}, \mathbf{R}_\mathbf{v}) \tag{16b}$$

where the subscript k refers to the discrete points in time t_k. The state, input and output vectors are related in a linear way via the matrices $\underline{\mathbf{A}} \in \mathbb{R}^{n_x \times n_x}$, $\underline{\mathbf{B}} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{n_y \times n_u}$ and $\mathbf{D} \in \mathbb{R}^{n_y \times n_u}$. Notice the small variations symbol δ has been omitted from the remainder of this paper to simplify notation.

3. HYBRID CONTROLLER SETUP

The term hybrid controller relates to the fact that four controllers, named K_I to K_{IV}, are active under different operating conditions governed by a switching mechanism.

The primary static objective of the wind turbine control system is to optimize power production for the given wind speed. Load reduction is a typical secondary objective, which shall however not be discussed further in this section. The primary objective for a given wind speed can be formulated as the constrained minimization of nonlinear quadratic cost function concerning generator power. An additional term concerning generator speed is included in the cost function, this term is only active when the first term concerning power is zero. The steady state constrained optimization problem for a given wind speed is

$$\min_{\Omega_g, \theta} (P_e - P_{nom})^2 + w(\Omega_g - \Omega_{g,nom})^2 \tag{17a}$$

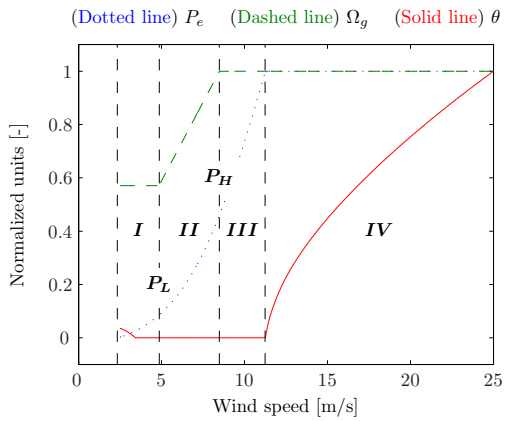


Figure 1. Sweep of wind speeds showing the steady state values of the primary variables of the wind turbine. Notice the pitch angle would normally have slightly different values for wind speeds below rated wind speed. The pitch angle in this case have been constrained and is thus different from its unconstrained value. This is however not expected to have significant influence on the performance of the controller.

Wind Energ. 2010; **00**:1–16 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

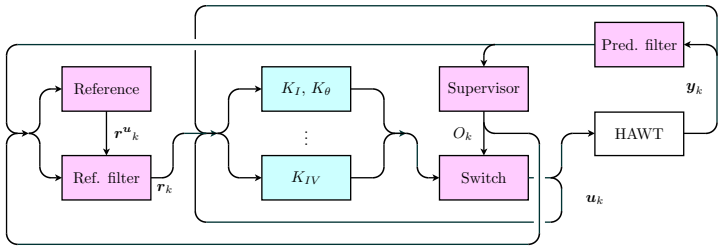


Figure 2. Setup of the hybrid controller. Regions I to III are only controlled by the generator torque controllers K_I to K_{III}. The pitch angle is kept constant by a separate controller K_θ in these regions. In region IV both pitch angle and generator torque are used in the same controller.

where

$$w = \begin{cases} 0 & \text{for} \quad P_e \neq P_{nom} \\ 1 & \text{for} \quad P_e = P_{nom} \end{cases}$$

subject to

$$\mathbf{0} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{17b}$$

$$\Omega_g \in [\Omega_{g,low}, \Omega_{g,nom}] \tag{17c}$$

$$\theta \in [\theta_{min}, \theta_{max}] \tag{17d}$$

where the equality constraint (17b) ensures steady state operation. The generator speed and pitch angle are limited to a certain ranges (17c) and (17d). The terms min/max indicate constraints to be respected, whereas low/nom indicate lower and upper set points that should be tracked. As explained later in this section the actual set point depends on the given wind speed: For low wind speed the low set point is tracked, for mid-range wind speeds the set point is between low and nom and for high wind speeds the set point is nom. Set points are not respected in the same way as constraints and e.g. generator speed is allowed to vary around its given set point, whereas e.g. the pitch angle cannot go beyond its min/max limits. The distinction between min/max and low/nom is maintained throughout this paper. There are also other constraints such as pitch rate and acceleration limits etc. but these velocity constraints are not active during the present steady state optimization sweep, as velocities and accelerations are zero at steady state. Above rated wind speeds, where the generator power is at its nominal value, there is no unique solution for the optimization problem. The pitch angle and generator speed could have any number of solutions as long as they are on the appropriate contour line of the C_P curve. The generator speed weight w ensures that for above rated wind speed there is a unique solution with generator speed at its nominal value.

The optimization gives the characteristic diagrams for generator power, generator speed and pitch angle versus wind speed as seen in Figure 1. Four regions of operation $O \in (I, \ldots, IV)$ also denoted $(O_I, \ldots, O_{IV})$ respectively, are derived from (17). The regions and their respective primary control objectives are:

- O_I - Optimize power. Generator speed at lower level. Pitch angle fixed at its minimum value.
- O_{II} - Optimize power. Variable generator speed, i.e. at global maximum of aerodynamic power coefficient C_P. Pitch angle fixed at its minimum value.
- O_{III} - Optimize power. Generator speed at nominal level. Pitch angle fixed at its minimum value.
- O_{IV} - Regulate power and generator speed to nominal levels. Variable pitch.

The resulting hybrid controller has a setup as shown in Figure 2. The pitch angle should should ideally be at its optimum in for a given wind speed in O_I-O_{III}. Instead, the pitch angle is chosen to be at its minimum value in these regions to simplify the control problem. The choice of fixating the pitch angle at lower wind speeds means that only the generator torque is used as a control variable in the controllers K_I to K_{III} . In these regions the pitch angle is controlled by a separate controller K_θ which simply aims to keep the pitch angle constant at the prescribed value. The constant pitch angle in O_I to O_{III} has little impact of the aerodynamic efficiency due to the flatness of the C_P curve. In O_{II} the aerodynamic power coefficient is assumed to be maximum

$$C_{P\,opt} = C_P(\theta_{opt}, \lambda_{opt}) \tag{18}$$

the tracking of the optimal aerodynamic coefficient in this region can be obtained by

$$P_{opt}(V) \approx P_{opt}(\Omega_g) = k_{opt}\Omega_g^3 \tag{19}$$

where

$$k_{opt} = \eta\tfrac{1}{2}\pi\rho R^5\lambda_{opt}^{-3}N_g^{-3}C_{P\,opt} \tag{20}$$

and where $V = \lambda_{opt}^{-1}\Omega_r R$ is presumed [16, 17]. One method for controlling the wind turbine in O_{II} is to determine either the desired optimal power or generator torque as function of generator speed as seen above. In this case, the combined mass moment of inertia of the rotor and generator will act as a low pass filter. The approach of tracking the generator power as function of generator speed can be referred to as $P\Omega$ control. The relationship can be reversed such that the generator speed is tracked as function of generator power (ΩP control)

$$\Omega_{g,opt}(V) \approx \Omega_{g,opt}(P_e) = \sqrt[3]{P_e k_{opt}^{-1}} \tag{21}$$

The tracking of generator speed could potentially lead to undesirable behavior of the closed loop since the control signal ($Q_{g,ref}$) is also a factor in the reference signal ($P_e = Q_g\Omega_g\eta$) only filtered by the relatively fast dynamics of the generator torque actuator and controller gains. Although the ΩP control could be potentially be problematic, it has been chosen in the present work to ensure a similar structure of the controller K_I,K_{II} and K_{III}. No major problems regarding the almost direct coupling between control and reference has been observed in the present work. If problems did occur an additional low pass filter could be inserted. In the other regions of operation, the set points have fixed values as listed in Table I.

The supervisor block in Figure 2 follows the switching conditions listed in Table II to determine which controller is active. The conditions S_{12} and S_{32} ensure switching from the fixed speed regions O_I and O_{III} to the variable speed region O_{II}. Apart from the condition that the supervisor should switch to O_{II} if the generator speed deviates from the fixed speed reference an additional condition concerning generator power is included in S_{12} and S_{32}. The assumption of an optimal power coefficient can be used to calculate the power levels at the critical wind speeds at the borders of region O_{II}

$$P_L = k_{opt}\Omega_{g,low}^3 \tag{22}$$

$$P_H = k_{opt}\Omega_{g,nom}^3 \tag{23}$$

The added generator power condition in S_{12} and S_{32} alleviates the problem of erratic switching between the fixed speed regions and the variable speed region. If the controller is already in the variable speed region and the generator speed exceeds its limits then S_{21} and S_{23} ensures switching to the fixed speed regions.

If the controller is in nominal speed region O_{III} and the generator power exceeds its nominal value then S_{34} switches to the nominal power and speed region O_{IV}. A shortcut condition S_{24} from the variable speed region switches directly from O_{II} to O_{IV} if extreme wind conditions occur and disturbs normal operation. The last condition S_{43} switches from O_{IV} to O_{III} if the pitch angle approaches its minimum limit and the generator power and speed are below their nominal values. The factors ϵ^- and ϵ^+ prohibits the switching mechanism from continuously switching between two neighboring operating regions, when the wind turbine operates on the boundaries between the respective regions. In this work ϵ^- and ϵ^+ have been chosen to be 0.99 and 1.01 respectively.

K_x	$\boldsymbol{y}^{\boldsymbol{r}}$	$\boldsymbol{r}^{(\boldsymbol{u})}$	$\boldsymbol{u}$
I	$[\Omega_g]$	$[\Omega_{g,low}]$	$[Q_{g,ref}]$
II	$[\Omega_g]$	$[\Omega_{g,opt}(P_e)]$	$[Q_{g,ref}]$
III	$[\Omega_g]$	$[\Omega_{g,nom}]$	$[Q_{g,ref}]$
θ	$[\theta]$	$[\theta_{opt}]$	$[\theta_{ref}]$
IV	$\begin{bmatrix}\Omega_g\\ P_e\end{bmatrix}$	$\begin{bmatrix}\Omega_{g,nom}\\ P_{nom}\end{bmatrix}$	$\begin{bmatrix}Q_{g,ref}\\ \theta_{ref}\end{bmatrix}$

Table I. Controlled outputs $\boldsymbol{y}^{\boldsymbol{r}}$, (undamped) set points $\boldsymbol{r}^{(\boldsymbol{u})}$ and control signals $\boldsymbol{u}$ for the underlying controllers K_x of the hybrid controller.

S	Switching conditions
12	$(\Omega_g \geq \Omega_{g,low}) \wedge (P_e \geq P_L)$
21	$(\Omega_g \leq \Omega_{g,low}\epsilon^-)$
23	$(\Omega_g \geq \Omega_{g,nom})$
24	$(P_e \geq P_{nom})$
32	$(\Omega_g \leq \Omega_{g,nom}\epsilon^-) \wedge (P_e \leq P_H\epsilon^-)$
34	$(P_e \geq P_{nom})$
43	$(\Omega_g < \Omega_{g,nom}\epsilon^-) \wedge (\dot{\Omega}_g < 0)$ $\wedge\, (P_e < P_{nom}\epsilon^-) \wedge (\theta < \theta_{min}\epsilon^+)$

Table II. Switching conditions S of supervisor block in hybrid controller. The first integer in S denotes the current operating region and the second integer indicates the operating region to be switched to.

Wind Energ. 2010; **00**:1–16 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

3.1. Reference filter

To avoid violent transitions between different operating regions, a filter is inserted between the individual controllers and the reference block. At times of switching, when $O_{k-1} \neq O_k$, the reference filter ensures that the reference sent to the controllers is initially the measured controlled output and that the reference sent to the controllers approaches the real reference at a rate specified by the reference filter. The reference to track $\boldsymbol{r}$ is damped by a first order filter

$$\boldsymbol{r} = \boldsymbol{y}^{\boldsymbol{r}}_{k_s} + \frac{1}{s\tau_r + 1}(\boldsymbol{r}^{\boldsymbol{u}} - \boldsymbol{y}^{\boldsymbol{r}}_{k_s}) \tag{24}$$

where $\boldsymbol{y}^{\boldsymbol{r}}_{k_s}$ is the measured value of the controlled variable at the time of switch and $\boldsymbol{r}^{\boldsymbol{u}}$ is the undamped reference. At the occurrence of a switch of operating region, the states of the filter are reset. In this work the time constant τ_r has been chosen to be 0.1 s.

3.2. Predictive filter

As it will be explained in the next section about predictive Kalman filters, the controllers rely not on the current measurement but solely on the previous one. This means that the Supervisor, Reference and Reference filter blocks in Figure 2 should also predict one sample ahead and determine which controller is active and what reference the active controller should track. A number of predictive filters could be used, including either of the Kalman filters of the controllers K_I to K_{IV}, in the present work the the state predictions from the estimator in K_{II} are used by the Supervisor, Reference and Reference filter blocks rather than the direct measurement.

4. MODEL PREDICTIVE CONTROL

The generic setup for the origin shifting controller is now explained. The controllers K_I to K_{IV} and K_θ are all set up as depicted in Figure 3. The implemented model predictive controller consists of two components: The first component ensures offset-free performance at steady state (Section 4.1) which is based on a general method that can also be used with other types of state space controllers such as pole placement and linear quadratic controllers. The second component of the controller is the constrained linear quadratic regulator (Section 4.2) which is similar to a standard LQ controller but has the ability handle hard and soft constraints.

4.1. Offset-free reference tracking in steady state, state estimation and origin shifting

Methods ensuring offset-free control such as the use of the disturbance model used in this paper or the error-integral-term used by a PI controller, handle the unmodeled effects caused by parametric uncertainties or unknown disturbances. Offset-free control ensures, for a constant reference $\boldsymbol{r} \in \mathbb{R}^{n_r}$, that the tracked outputs $\boldsymbol{y}^{\boldsymbol{r}} \in \mathbb{R}^{n_r}$ of the plant have zero offset with regards to $\boldsymbol{r}$ in steady state. In time-discrete formulation, offset-free control can be written as

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} (\boldsymbol{y}^{\boldsymbol{r}}{}_k - \boldsymbol{r}_k) = 0 \tag{25}$$

The origin shifting controller presented in this section aims to track the reference or set point signal $\boldsymbol{r}$

$$\boldsymbol{r}_k = \boldsymbol{y}^{\boldsymbol{r}}{}_k, \quad \boldsymbol{y}^{\boldsymbol{r}}{}_k \equiv \mathbf{H}\boldsymbol{y}_k \tag{26}$$

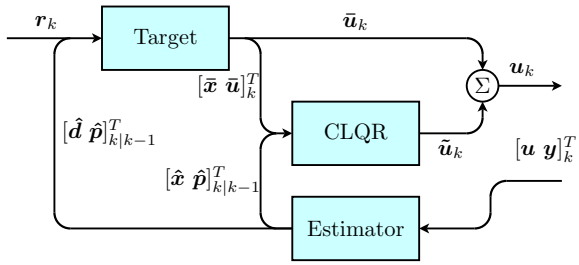


Figure 3. Origin shifting model predictive controller setup. Notice that the control signal given by the controller is not directly fed back to the estimator. This is because only the control signal of the active controller in the hybrid controller is fed back to the estimator.

where $\mathbf{H} \in \mathbb{R}^{n_r \times n_y}$ gives a linear combination or subset of the outputs and $\boldsymbol{y^r}$ is called the controlled or tracked outputs. In the case of a wind turbine, this could be desired generator speed and desired generator power.

If some sort of offset-free method is not used, then (25) cannot be ensured. To ensure offset-free performance, the control design model (16) can be augmented with a disturbance model [18, 19], which is used to compensate for actual disturbances and for any plant/model mismatch.

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} \underline{\mathbf{A}} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\underline{\hat{\mathbf{A}}}} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_{k} + \underbrace{\begin{bmatrix} \underline{\mathbf{B}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\underline{\hat{\mathbf{B}}}} \boldsymbol{u}_k + \begin{bmatrix} \boldsymbol{w_x} \\ \boldsymbol{w_d} \\ \boldsymbol{w_p} \end{bmatrix}_{k} \tag{27a}$$

$$\boldsymbol{y}_k = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{F} \end{bmatrix}}_{\hat{\mathbf{C}}} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_{k} + \mathbf{D}\boldsymbol{u}_k + \boldsymbol{v}_k \tag{27b}$$

where

$$\boldsymbol{v} \in N(\mathbf{0}, \mathbf{R_y}) \text{ and } \begin{bmatrix} \boldsymbol{w_x} \\ \boldsymbol{w_d} \\ \boldsymbol{w_p} \end{bmatrix} \in N(\mathbf{0}, \mathbf{R_{xdp}})$$

and where $\boldsymbol{d} \in \mathbb{R}^{n_d}$ are the state disturbances and $\boldsymbol{p} \in \mathbb{R}^{n_p}$ are the output disturbances and where $\mathbf{R_{xdp}} \in \mathbb{R}^{(n_x+n_d+n_p)\times(n_x+n_d+n_p)}$ is variance of the states and disturbances and $\mathbf{R_y} \in \mathbb{R}^{n_y \times n_y}$ is the variance of the output. The clear distinction between output and state disturbances eases the design of an estimator with regards to tuning as discussed later in this section.

Because the disturbances cannot be measured they have to be estimated. A discrete time predictive Kalman filter is designed to estimate states and disturbances. The predictive Kalman filter, which only bases its estimation on the previous sample, is chosen instead of the ordinary Kalman filter, which bases its estimation on the previous and current sample. For a full state feedback control law, with feedback gain $\mathbf{K}$, the difference between using either a predictive or an ordinary Kalman filter would be $\boldsymbol{u}_k = -\mathbf{K}\hat{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{u}_k = -\mathbf{K}\hat{\boldsymbol{x}}_{k|k}$, respectively. The choice of the predictive Kalman filter is necessary since some measurements include a direct term from input to output, e.g. pitch acceleration and generator torque rate, and the problem of causality would arise if an ordinary Kalman filter was implemented. Furthermore the predictive Kalman filter gives the computationally expensive MPC algorithm an entire sample time period to compute the control signal.

The estimated states and disturbances are denoted $[\hat{\boldsymbol{x}}\ \hat{\boldsymbol{d}}\ \hat{\boldsymbol{p}}]^T_{k|k-1}$, where $k|k-1$ means the estimation at time k given by the information at time $k-1$. The predictive Kalman filter is given by

$$\begin{aligned} \begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{d}} \\ \hat{\boldsymbol{p}} \end{bmatrix}_{k+1|k} &= \underline{\hat{\mathbf{A}}} \begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{d}} \\ \hat{\boldsymbol{p}} \end{bmatrix}_{k|k-1} + \underline{\hat{\mathbf{B}}}\boldsymbol{u}_k \\ &\quad + \mathbf{L}[\boldsymbol{y}_k - \hat{\mathbf{C}} \begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{d}} \\ \hat{\boldsymbol{p}} \end{bmatrix}_{k|k-1} - \mathbf{D}\boldsymbol{u}_k] \end{aligned} \tag{28}$$

where the predictive Kalman gain $\mathbf{L} \in \mathbb{R}^{(n_x+n_d+n_p)\times n_y}$ is determined by the discrete time steady state algebraic Riccati equation

$$\mathbf{L} = \underline{\hat{\mathbf{A}}}\mathbf{P}\hat{\mathbf{C}}^T[\hat{\mathbf{C}}\mathbf{P}\hat{\mathbf{C}}^T + \mathbf{R_y}]^{-1} \tag{29a}$$

$$\mathbf{P} = \mathbf{R_{xdp}} + \underline{\hat{\mathbf{A}}}\mathbf{P}\underline{\hat{\mathbf{A}}}^T - \mathbf{L}\hat{\mathbf{C}}\mathbf{P}\underline{\hat{\mathbf{A}}}^T \tag{29b}$$

The state and output variance can be considered to be given by normal distributed white noise whereas the variance properties of the disturbances could either be given by normal distributed white noise or by the deterministic discrepancies between the nonlinear plant and the linear model. For very nonlinear plants such as a wind turbine with its nonlinear aerodynamics the nonlinearities will dominate the performance of the disturbance estimation and effort should be concentrated on simple tuning rules for the estimator. The tuning of this estimator can be conducted via 4 scalars

DOI: 10.1002/we
Prepared using weauth.cls

(r_x, r_d, r_p, r_y)

$$\mathbf{R_{xpd}} = \text{diag}(r_x\mathbf{I}\ r_d\mathbf{I}\ r_p\mathbf{I}) \text{ and } \mathbf{R_y} = r_y\mathbf{I}$$

If the stochastic information about the model is disregarded under the assumption that the deterministic nonlinearities are dominating the performance of the disturbance estimation then the following tuning guidelines can be used: For $r_p/(r_x, r_d, r_y) \to \infty$ the disturbances are tracked instantly and for $r_p/(r_x, r_d, r_y) \to 1$ the disturbances are tracked at an increasingly slower rate. This simple tuning is possible due to structure of the disturbance model. If a more complicated model where stochastic properties of states and outputs are considered and the disturbance model structure is different than the one suggested here then an entirely different approach of tuning should be chosen.

The disturbance model can be structured in many ways but from a practical perspective it should be simple to create. A simple design rule used in this work is to include a state disturbance for each measured state, and an output disturbance for each measured output that is not simply a state but a function of several states and inputs.

In steady state, when $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and the state and output noise are given by their mean values, the system equations are given by

$$\boldsymbol{x} = \underline{\mathbf{A}}\boldsymbol{x} + \underline{\mathbf{B}}\boldsymbol{u} + \mathbf{E}\boldsymbol{d} \tag{30a}$$

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u} + \mathbf{F}\boldsymbol{p} \tag{30b}$$

Combining (30) and (26) gives the linear system of equations to be solved

$$\begin{bmatrix} \underline{\mathbf{A}} - \mathbf{I} & \underline{\mathbf{B}} \\ \mathbf{HC} & \mathbf{HD} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}} \\ \bar{\boldsymbol{u}} \end{bmatrix}_k = \begin{bmatrix} -\mathbf{E}\hat{\boldsymbol{d}} \\ \boldsymbol{r} - \mathbf{HF}\hat{\boldsymbol{p}} \end{bmatrix}_k \tag{31}$$

which gives the steady state target values $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$ that results in the desired references to be reached without offset in spite of external disturbances and model/plant mismatch. Applying $\bar{\boldsymbol{u}}$ to the plant will under the assumption of constant disturbances and a time-invariant plant ensure offset-free reference tracking. Relying solely on this approach will result in slow convergence between controlled outputs and references. A dynamic controller, which could be e.g. based on LQ- or H_∞-theory etc., is used to drive the states toward their target values at a faster pace. Origin shifted variables are introduced

$$\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \bar{\boldsymbol{x}} \text{ and } \tilde{\boldsymbol{u}} = \boldsymbol{u} - \bar{\boldsymbol{u}}$$

and the role of dynamic controller is to drive the origin shifted states toward zero $((\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}) \to 0)$ at a rate specified by the tuning of the dynamic controller.

A set of conditions regarding number of controlled output versus inputs, number of augmented disturbances etc. have to be fulfilled to ensure offset-free control, [19] can be consulted for an elaboration on these conditions.

4.2. Constrained linear quadratic regulator

The constrained linear quadratic regulator (CLQR) entails the computation of the control signal within a prediction horizon in the range $i = (0, \ldots, N)$ where k is the global time index and i is the local receding time index. At time k the prediction horizon ranges from k to $k + N$. The CLQR is formulated as a dual mode horizon where the first part, i.e $i = (0, \ldots, N)$, is considered constrained. In the second horizon, i.e. $i = (N + 1, \ldots, \infty)$, it is assumed that the plant has reached a state where the unconstrained solution is feasible [20].

New variables are introduced: An optimization variable $\boldsymbol{z} \in \mathbb{R}^{n_z}$ which contains the quantities that should be driven towards their offset-free target values. Hard and soft constraints are also introduced, respectively denoted $\boldsymbol{h} \in \mathbb{R}^{n_h}$ and $\boldsymbol{s} \in \mathbb{R}^{n_s}$

$$\boldsymbol{z}_k = \mathbf{C_z}\boldsymbol{x}_k + \mathbf{D_z}\boldsymbol{u}_k + \mathbf{F_z}\boldsymbol{p}_k \tag{32}$$

$$\boldsymbol{h}_k = \mathbf{C_h}\boldsymbol{x}_k + \mathbf{D_h}\boldsymbol{u}_k + \mathbf{F_h}\boldsymbol{p}_k \tag{33}$$

$$\boldsymbol{s}_k = \mathbf{C_s}\boldsymbol{x}_k + \mathbf{D_s}\boldsymbol{u}_k + \mathbf{F_s}\boldsymbol{p}_k \tag{34}$$

where $\mathbf{C_z} \in \mathbb{R}^{n_z \times n_x}$, $\mathbf{D_z} \in \mathbb{R}^{n_z \times n_u}$, $\mathbf{C_h} \in \mathbb{R}^{n_h \times n_x}$, $\mathbf{D_h} \in \mathbb{R}^{n_h \times n_u}$ $\mathbf{C_s} \in \mathbb{R}^{n_s \times n_x}$ and $\mathbf{D_s} \in \mathbb{R}^{n_s \times n_u}$ describe the linear relationship between the new variables and $\boldsymbol{x}$ and $\boldsymbol{u}$. The matrices $\mathbf{F_p} \in \mathbb{R}^{n_z \times n_p}$, $\mathbf{F_h} \in \mathbb{R}^{n_h \times n_p}$ and $\mathbf{F_s} \in \mathbb{R}^{n_s \times n_p}$ describe the relationship between the new variables and output disturbances. If for instance the constrained variables are not measured, then offset-free honoring of the constraints cannot be guaranteed. In the case of the wind turbine controller the optimization, hard constraint and soft constraint vectors, respectively are

$$\boldsymbol{z} = \begin{bmatrix} \Omega_r\ \Omega_g\ \phi\ \dot{\psi}_t\ \theta\ \dot{\theta}\ \ddot{\theta}\ Q_g\ \dot{Q}_g\ P_e \end{bmatrix}^T \tag{35}$$

$$\boldsymbol{h} = \begin{bmatrix} \ddot{\theta}\ \dot{Q}_g \end{bmatrix}^T \tag{36}$$

$$\boldsymbol{s} = \begin{bmatrix} \theta\ \dot{\theta}\ Q_g\ \Omega_g\ P_e \end{bmatrix}^T \tag{37}$$

The sensitivity matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C_z}$, $\mathbf{D_z}$ etc. can be determined analytically or through numerical differentiation. It is considered beyond the scope of this paper to provide the actual numbers in the sensitivity matrices and the reader is requested to consult [8] for the parameters used by the model. The resulting optimization problem in general terms can be written as

$$\min_{\vec{\boldsymbol{x}}_k,\vec{\boldsymbol{u}}_k,\vec{\boldsymbol{\sigma}}_k} \underbrace{\sum_{i=k}^{k+N} \tilde{\boldsymbol{z}}_{i|k}^T \mathbf{W} \tilde{\boldsymbol{z}}_{i|k} + \tilde{\boldsymbol{\sigma}}_{i|k}^T \boldsymbol{\Xi} \tilde{\boldsymbol{\sigma}}_{i|k}}_{\text{1st mode}} + \underbrace{\sum_{i=k+N+1}^{\infty} \tilde{\boldsymbol{z}}_{i|k}^T \mathbf{W} \tilde{\boldsymbol{z}}_{i|k}}_{\text{2nd mode}} \tag{38a}$$

subject to

$$\tilde{\boldsymbol{x}}_{k|k} = \tilde{\boldsymbol{x}}_k \tag{38b}$$

$$\tilde{\boldsymbol{x}}_{i+1|k} = \underline{\mathbf{A}} \tilde{\boldsymbol{x}}_{i|k} + \underline{\mathbf{B}} \tilde{\boldsymbol{u}}_{i|k} \qquad i = (k, k+1, \ldots, \infty) \tag{38c}$$

$$\tilde{\boldsymbol{h}}_{i|k} \leq \boldsymbol{h_{lim}} - \bar{\boldsymbol{h}}_k \qquad i = (k, k+1, \ldots, k+N) \tag{38d}$$

$$\tilde{\boldsymbol{s}}_{i|k} \leq \boldsymbol{s_{lim}} - \bar{\boldsymbol{s}}_k + \boldsymbol{\sigma}_{i|k} \qquad i = (k, k+1, \ldots, k+N) \tag{38e}$$

$$\boldsymbol{\sigma}_{i|k} \geq \mathbf{0} \qquad i = (k, k+1, \ldots, k+N) \tag{38f}$$

where $\vec{\boldsymbol{x}}_k = [\tilde{\boldsymbol{x}}_{k+1|k}, \ldots, \tilde{\boldsymbol{x}}_{k+N+1|k}]^T$, $\vec{\boldsymbol{u}}_k = [\tilde{\boldsymbol{u}}_{k|k}, \ldots, \tilde{\boldsymbol{u}}_{k+N|k}]^T$ and $\vec{\boldsymbol{\sigma}}_k = [\tilde{\boldsymbol{\sigma}}_{k|k}, \ldots, \tilde{\boldsymbol{\sigma}}_{k+N|k}]^T$. The primary component of the cost function (38a) is the weight matrix $\mathbf{W} \in \mathbb{R}^{n_z \times n_z}$ which determines how much each of the optimization variables $\tilde{\boldsymbol{z}}$ should be minimized. The optimization problem is subject to the initial condition equality (38b) and the state transition equation (38c). The first mode of the prediction horizon is subjected to constraints. In the second mode of the prediction horizon, the system is assumed to be sufficiently far away from any constraints. The inequality constraints of the optimization problem is divided into the hard inequality constraints (38d) and the soft inequality (38e). Hard constraints cannot be violated and if they are violated, the constrained optimization problem becomes infeasible and no proper solution can be found [21]. To remedy the potential infeasibility soft constraints are introduced. The violation of soft constraints is given by the slack variable $\boldsymbol{\sigma} \in \mathbb{R}^{n_s}$ (38f) which is minimized with the use of the second component of the cost function (38a) $\boldsymbol{\Xi} \in \mathbb{R}^{n_s \times n_s}$.

The unconstrained second mode minimization can be reduced to a terminal cost $\mathbf{S} \in \mathbb{R}^{n_x \times n_x}$ in the constrained problem (38)

$$\min \sum_{i=k+N+1}^{\infty} \tilde{\boldsymbol{z}}_{i|k}^T \mathbf{W} \tilde{\boldsymbol{z}}_{i|k} = \tilde{\boldsymbol{x}}_{k+N+1|k}^T \mathbf{S} \tilde{\boldsymbol{x}}_{k+N+1|k} \tag{39}$$

which gives infinite horizon properties and acts stabilizing [20]. The terminal cost $\mathbf{S}$ and feedback gain $\mathbf{K}$ are determined by the time-discrete steady state algebraic Riccati equation

$$\mathbf{K} = [\underline{\mathbf{B}}^T \mathbf{S} \underline{\mathbf{B}} + \mathbf{D_z}^T \mathbf{W} \mathbf{D_z}]^{-1} [\underline{\mathbf{B}}^T \mathbf{S} \underline{\mathbf{A}} + \mathbf{D_z} \mathbf{W} \mathbf{C_z}^T] \tag{40a}$$

$$\mathbf{S} = \mathbf{C_z}^T \mathbf{W} \mathbf{C_z} + \underline{\mathbf{A}}^T \mathbf{S} \underline{\mathbf{A}} - [\underline{\mathbf{A}}^T \mathbf{S} \underline{\mathbf{B}} + \mathbf{C_z}^T \mathbf{W} \mathbf{D_z}] \mathbf{K} \tag{40b}$$

The first mode of (38) with the appended terminal cost can be formulated as a quadratic programming (QP) problem, which can be solved using either interior-point or active-set algorithms [22]. From the computed sequence of control signals only the first signal $\tilde{\boldsymbol{u}}_k$ is actuated.

5. NUMERICAL EXAMPLE: MPC AND PI ON NREL 5MW REFERENCE WIND TURBINE

In this section the proposed MPC controller design is tested on the NREL 5MW reference wind turbine [8] in the aero-servo-elastic code HAWC2 [23] and compared to a more classical PI-based controller [8]. The pitch and generator actuators have been implemented as external dll's coupled to HAWC2. They are modeled as first and second order filters (13) and (14), with hard coded constraints on position and rate.

The PI controller has been implemented with gain scheduling of the pitch controller gains to accommodate for the nonlinear relationship between pitch angle and the aerodynamic power efficiency. The implemented PI controller is similar

DOI: 10.1002/we
Prepared using weauth.cls

to the PI controller presented in [8], but has been modified with a different integrator anti-windup scheme to accommodate saturations in pitch rate. Without this modification the closed-loop system could be unstable during extreme events, if pitch rate is saturated over an extended period of time leading to integrator windup. The modification implies that the PI controller is only able to use collective pitch, but as collective pitch is the only configuration being tested in the present work this has no influence on the results. The integrator anti-windup scheme only considers the output of the PI controller and not the actual output of the actuators.

Where the PI controller ignores the actuator dynamics, the MPC includes them in the present work. The speed of the actuator dynamics are important in the considerations of whether or not to include the actuator dynamics in the control design model. If the actuator dynamics are significantly faster than the wind turbine turbine dynamics and faster than the sample time of the controller, it might be better to omit the actuator dynamics from the control design model of the MPC and instead impose constraints directly on the control signal and its derivatives similar to the implementation of the PI. If, on the other hand, the dynamics of the actuators are slow enough to have an significant influence on the closed-loop dynamics, the actuator dynamics should be included in the control design model. An inclusion of the actuator dynamics in the control design model enables the MPC to control and constrain the output of the actuators, where the PI disregards the actuator dynamics and lags a bit behind the MPC. The potential benefit of including the actuator dynamics in the control design model increases for slower actuators.

In addition to the physical constraints of the actuators, artificial constraints have also been included in the model predictive controller. The electrical power of the generator is not allowed to increase above 5 percent of rated power according to an added ”semi-heavily” penalized soft constraint. It means that this constraint should not be violated unless even harder constraints are in danger of being violated. An even softer constraint specifying that the generator speed is not allowed to exceed 10 percent of its rated value is also included in the MPC. The generator speed constraint cannot be penalized as hard since situations might occur, where it is impossible to honor the generator speed constraint due to the large inertia of the drive-train, and the control problem would be rendered infeasible.

Four test cases are presented: Test case 1 is a sequence of wind speed steps, going from 4 m/s to 20 m/s and back down to 4 m/s gain in steps of 2 m/s. Test case 2 has a turbulent wind field based on the model of Mann [10] with turbulence intensity 0.16 and a power law wind shear with a 0.2 shear exponent. The mean wind speed ramps from 4 m/s to 16 m/s and back to 4 m/s. Test cases 1 and 2 are included to evaluate the performance of the hybrid controller and its ability to switch between the different regions of operation. Test cases 3 and 4 have mean wind speed of 12 m/s where an extreme operating gust occurs, as specified by IEC for a class I_A wind turbine [11]. In test case 3, the actuators have normal constraint limits. In test case 4, the pitch rate limit has been reduced significantly to test the constraint handling capabilities of both controllers.

In the following, results for the test cases are depicted in Figures 4 to 8. The y-axis for a given variable in the different figures has the same upper and lower limits, when possible, but for a given case, alternative limits have been used when thought better for the sake of clarity.

The results for test case 1 are shown in Figure 4. The two types of controllers show different behavior in certain areas. The gain scheduling of the PI ensures that closed-loop behavior above rated wind speed is similar over a wide range of wind speeds, whereas the MPC is linearized at 15 m/s and suffers from this simplification, when operating far away from the linearization point. Some sort of gain-scheduling scheme is required to overcome this problem. None the less, the MPC controller linearized at 15 m/s is able to work over large wind speed range. The periodic disturbances of wind shear and tower shadow are seen in the actuator states of the MPC controller. The MPC controller could be tuned differently to ignore these periodic disturbances instead of trying to reject them, which would lead to less control action, if desired.

Figure 5 depicts the results for case 2, it is seen that both controllers are able to operate in the full wind speed range and switch between regions of operation. The generator torque rate of the PI controller has a sudden transient at approx 510 s. where the controller switches from full load to partial load operation. The reference switching filters of the MPC controller prevent such transients and a smoother transition between different regions of operation is observed for the MPC controller.

Results for test cases 3 and 4 are shown in Figures 6, 7 and 8. As the generator speed exceeds its limit of 10 percent above rated speed, the generator torque is increased in an attempt to slow down the over-speeding wind turbine. The generator power is however only allowed to exceed 5 percent of its nominal power and the generator torque is controlled accordingly. The generator torque changes quite sudden to accommodate the attempt to slow the over-speeding wind turbine, especially in test case 4 seen in Figure 7. Perhaps this sudden change in generator torque is not good for real world application but in the given example it enables a reduction of the generator speed compared to the PI controller.

The benefit of including the actuator dynamics in the control design model becomes apparent in Figures 7 and 8, where the pitch reference rate of the PI is constrained. The MPC has no constraints directly on the pitch reference rate and small overshoots at 11 s and 14 s for the MPC can be observed in Figure 8. The small overshoots enables the MPC to reach the pitch rate limit at a faster rate than the PI, while still honoring the pitch rate limit, as show in Figure 7 where the pitch rate of MPC at 11 s and 14 s approaches it limit at a sharper curve than prescribed by the first order dynamics of the pitch actuator. The slower the dynamics of the actuator, the higher the benefit by including the actuator dynamics.

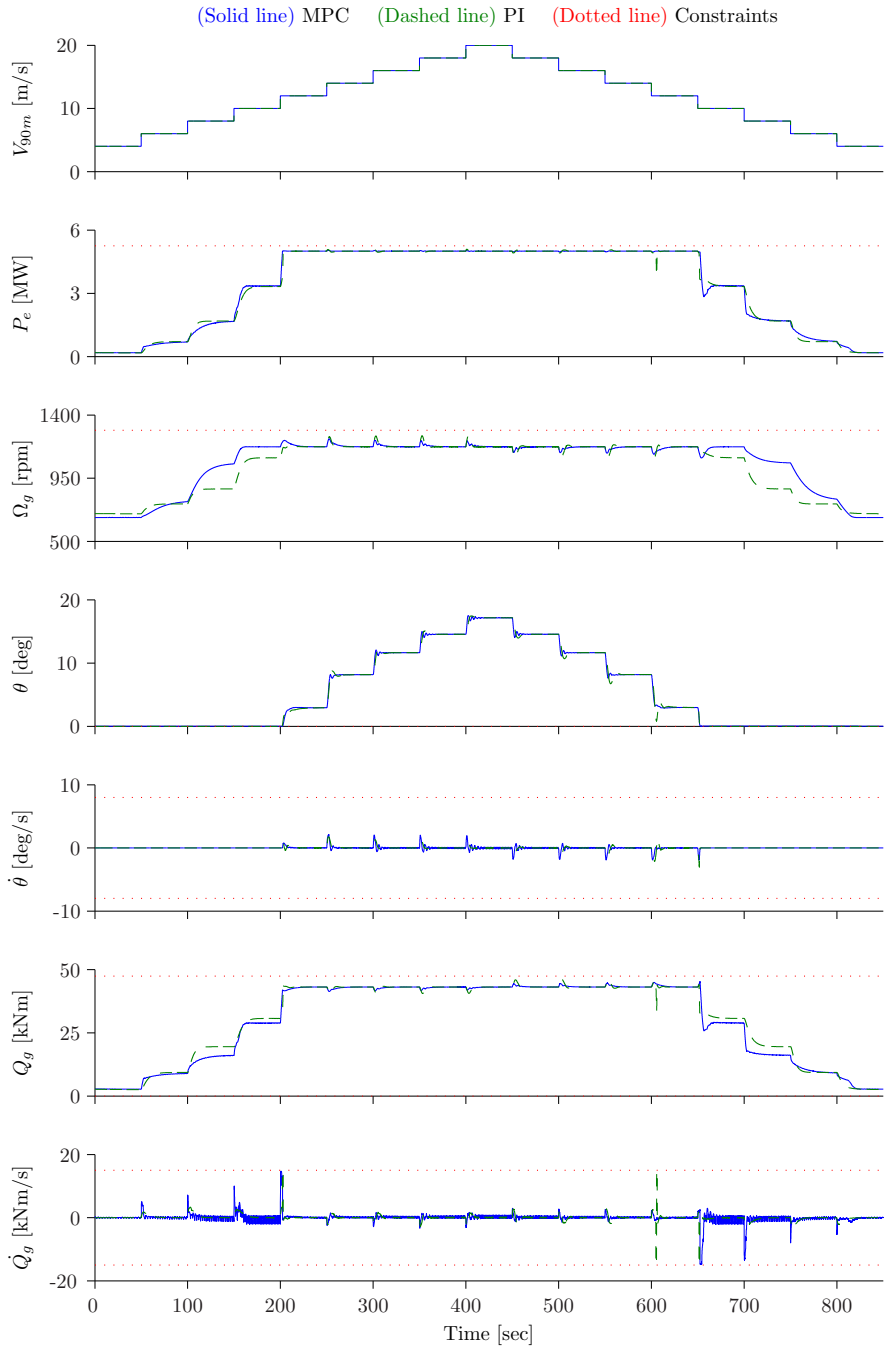


Figure 4. Test case 1 with a sequence wind steps in 2 m/s increments and decrements respectively.

Wind Energ. 2010; **00**:1–16 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

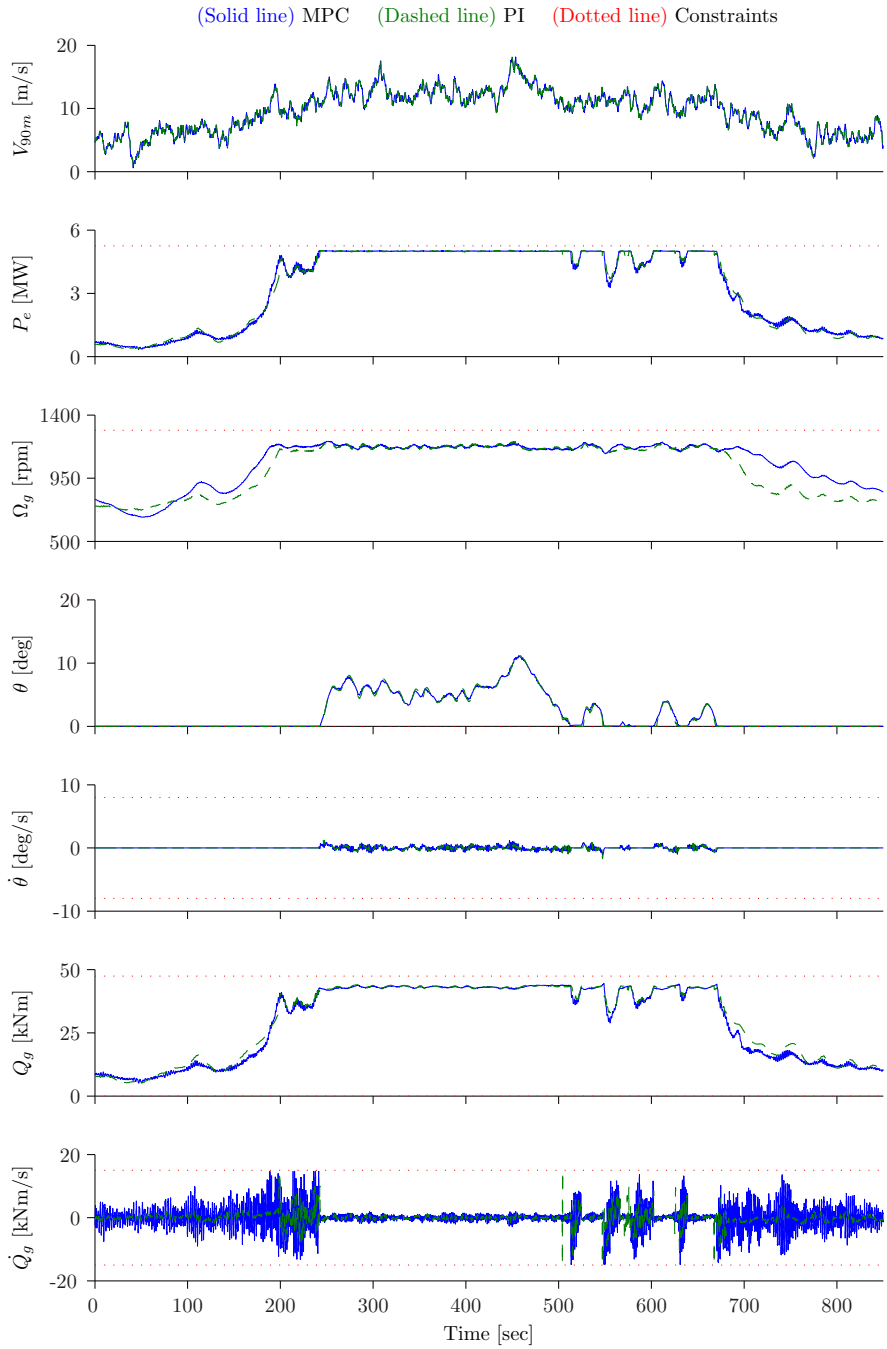


Figure 5. Test case 2 with turbulent wind. Mean wind speed ramps from 4 m/s to 16 m/s and back down to 4 m/s.

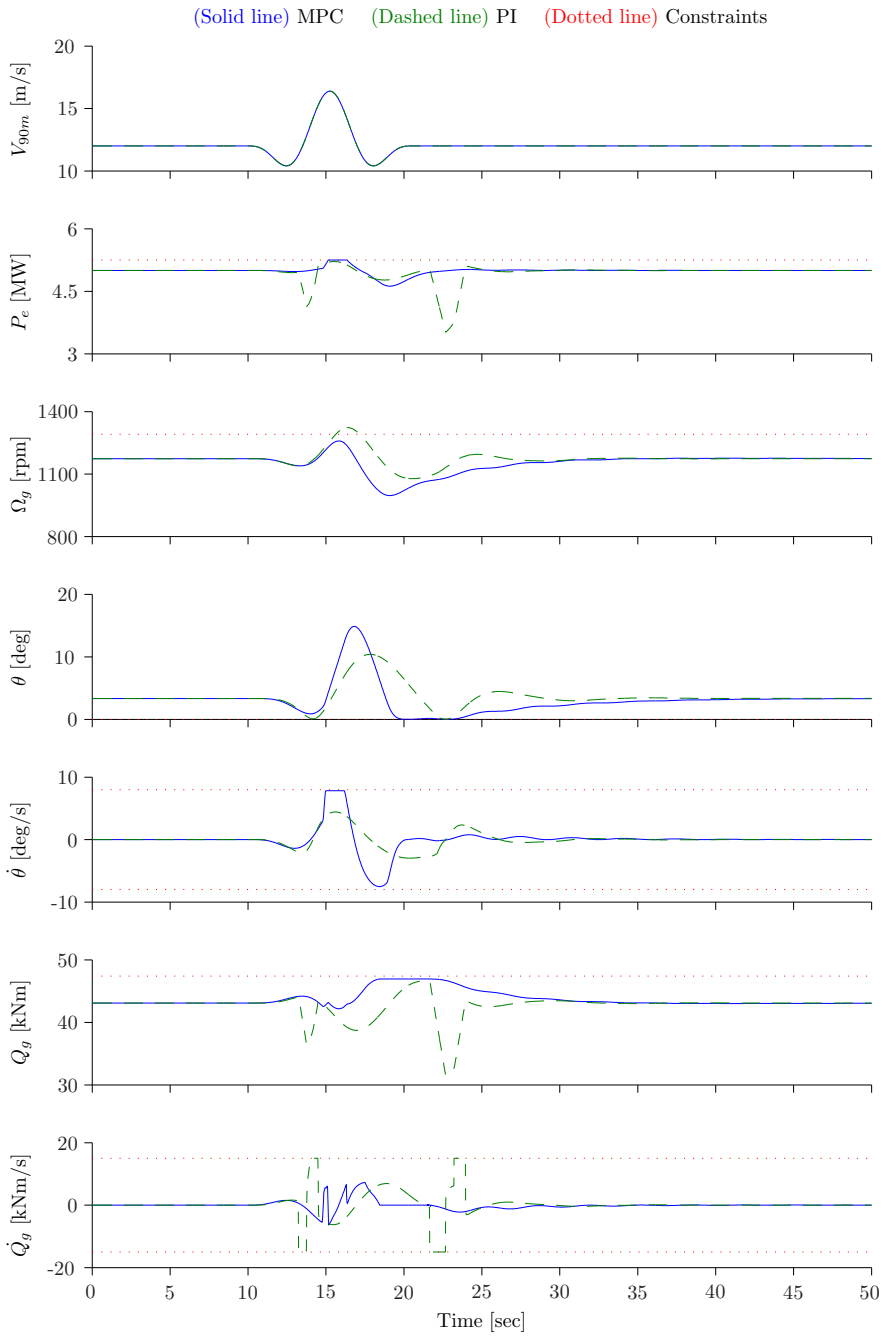


Figure 6. Test case 3 with extreme operating gust at a mean wind speed of 12 m/s. Tower shadow and wind shear has been disabled to achieve clean results. Pitch rate is constrained to 8 deg/s and generator torque rate is constrained to 15 kNm/s.

Wind Energ. 2010; **00**:1–16 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

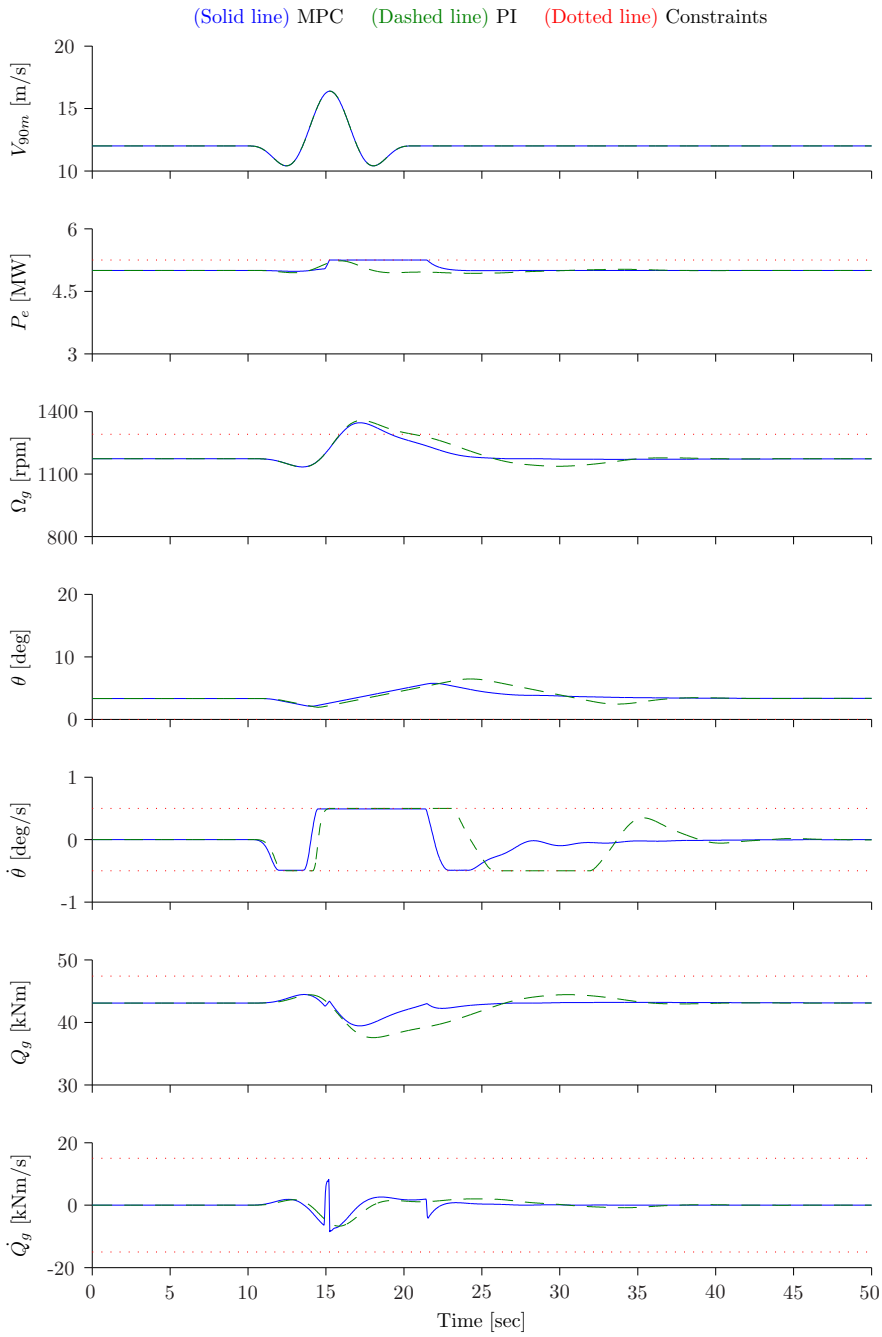


Figure 7. Test case 4 with extreme operating gust at a mean wind speed of 12 m/s. Tower shadow and wind shear has been disabled to achieve clean results. The pitch rate constraint have been drastically reduced from 8 deg/s to 0.5 deg/s to challenge the controllers.

6. CONCLUSION

In this paper the model predictive control within a hybrid controller framework is presented. A model describing the key elements of wind turbine dynamics have been presented, thus enabling model-based control methods. It has been shown that the hybrid controller setup handles changing wind speeds and operating conditions. Simulations show that model predictive control ensures that no hard constraints are violated and that soft constraints are attempted to be honored when possible. Generator speed and -power have been included as artificial soft constraints to demonstrate the capabilities of prioritizing which components should be attempted spared during extreme events. It has furthermore been demonstrated that the inclusion of actuator dynamics in the control design model is advantageous as it exploits the full potential of the actuators, an advantage that increases for slower actuator dynamics.

REFERENCES

1. Boukhezzar B, Siguerdidjane H, Hand M. Nonlinear control of variable-speed wind turbines for generator torque limiting and power optimization. *Transactions of the ASME. Journal of Solar Energy Engineering* 2006; **128**(4):516–30.
2. Wright AD, Balas MJ. Design of state-space-based control algorithms for wind turbine speed regulation. *Wind Energy and Journal of Solar Energy Engineering, Transactions of the ASME* 2003; **125 (4)**:386–395.
3. Ostergaard K, Stoustrup J, Brath P. Linear parameter varying control of wind turbines covering both partial load and full load conditions. *International Journal of Robust and Nonlinear Control* 2009; **19**(1):92–116.
4. Kanev S. Exploring the limits in individual pitch control. *Scientific Proceedings of the European Wind Energy Conference 2009*, Marseille, France, 2009.
5. Santos R. Damage mitigating control for wind turbines. PhD Thesis, University of Colorado at Boulder, United States – Colorado 2007.
6. Hammerum K, Brath P, Poulsen NK. A fatigue approach to wind turbine control. *Journal of Physics: Conference Series* 2007; **75**(1):012 081.
7. Trainelli L, Sirchi W, Savini B, Croce A, Bottasso CL. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody System Dynamics* 2006; **16**(3):291–308.
8. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-mw reference wind turbine for offshore system development. *Technical Report NREL/TP-500-38060*, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401-3393 February 2009.
9. Mann J. The spatial structure of neutral atmospheric surface-layer turbulence. *Journal of Fluid Mechanics* 1994; **273**:141–68.
10. Mann J. Wind field simulation. *Probabilistic Engineering Mechanics* 1998; **13**(4):269–282.
11. IEC/TC88. *IEC 61400-1 Ed.3: Wind turbines - Part 1: Design requirements*. International Electrotechnical Commission (IEC) 8 2005.
12. Henriksen LC, Poulsen NK. Energy-based nonlinear control of hydraulically actuated pitch-servo systems. *Proceedings of the European Wind Energy Conference 2009*, Marseille, France, 2009.
13. Hansen MH, Kallesøe BS. Servo-elastic dynamics of a hydraulic actuator pitching a blade with large deflections. *Journal of Physics: Conference Series* 2007; **75**(1):012 077.
14. Larsen T, Hansen M, Iov F. Generator dynamics in aeroelastic analysis and simulations. *Risø-R 1395*, Risø National Laboratory 2003.
15. Franklin GF, Powell JD, Workman M. *Digital Control of Dynamic Systems*. 2nd edn., Addison-Wesley, 1990.
16. Bossanyi E. Wind turbine control for load reduction. *Wind Energy* 2003; **6**(3):229–244.
17. Burton T, Sharpe D, Jenkins N, Bossanyi E. *Wind Energy Handbook*. John Wiley And Sons Ltd, 2001.
18. Muske KR, Badgwell TA. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control* 2002; **12**(5):617–632.
19. Pannocchia G, Rawlings JB. Disturbance models for offset-free model-predictive control. *AIChE Journal* 2003; **49**(2):426–437.
20. Scokaert P, Rawlings J. Constrained linear quadratic regulation. *Automatic Control, IEEE Transactions on* 1998; **43**(8):1163 –1169.
21. Scokaert PO, Rawlings JB. Feasibility issues in linear model predictive control. *AIChE Journal* 1999; **45**(8):1649–1659.
22. Nocedal J, Wright SJ. *Numerical Optimization*. 2nd edn., Springer, 2006.
23. Larsen TJ, Hansen AM. How 2 HAWC2, the user's manual. *Technical Report Risø-R-1597(ver. 3-1)(EN)*, Risø National Laboratory 2007.

DOI: 10.1002/we
Prepared using weauth.cls

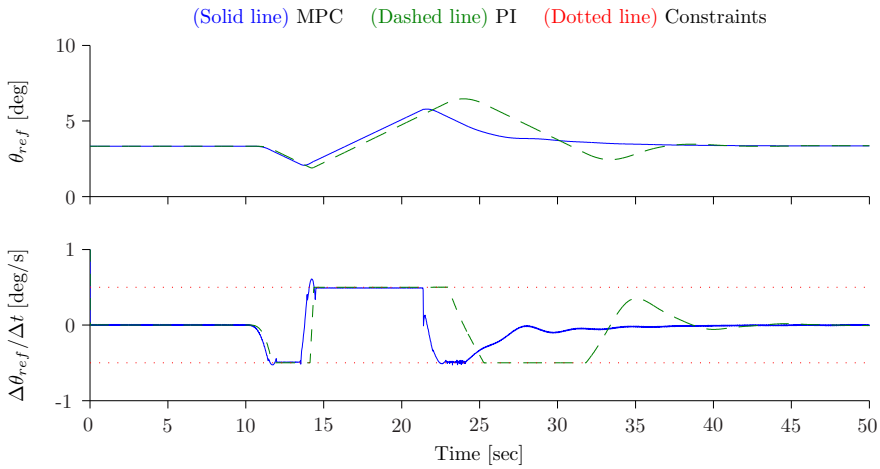


Figure 8. Test case 4 with extreme operating gust at a mean wind speed of 12 m/s. Pitch reference and pitch reference rate signals.

WIND ENERGY
Wind Energ. 2010; **00**:1–15
DOI: 10.1002/we

RESEARCH ARTICLE

Relinearized Model Predictive Control of a Floating Wind Turbine

L. C. Henriksen [1], M. H. Hansen [1], N. K. Poulsen [2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark
[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ABSTRACT

A continuously relinearized model-based control and state estimation algorithm applied to the control of a floating wind turbine is presented in this paper. The control algorithm covers both partial and full load operating regions. The performance of the presented controller is compared to a proportional-integral controller in three different simulations with mean speeds of 8, 12 and 16 m/s, respectively. Simulations show that the presented controller is able to switch between partial and full load operation in a smooth manner and that fatigue load reductions are achieved in a number of key components while reducing overall blade pitch activity compared to the benchmark proportional-integral controller.

KEYWORDS

model predictive control; constraint handling; floating horizontal axis wind turbine

Correspondence
L. C. Henriksen, Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark.
E-mail: larh@risoe.dtu.dk

Received . . .

1. INTRODUCTION

As an alternative to ground fixed offshore wind turbines, floating wind turbines offer a more versatile placement enabling further deployment of renewable energy sources. In Ref. [1] several concepts are presented, one of these concepts, the spar buoy also proposed by Ref. [2] is investigated in this work. Some of the challenges caused by the added degrees of freedom compared to a fixed wind turbine has previously been investigated by Ref. [3] , where it is concluded that the closed-loop frequency of the generator-speed-blade-pitch-loop should be below the tower pitch frequency to avoid unstable control performance. Ref. [4] use a state estimator to filter the disturbances caused by the tower pitch motion. Ref. [5] , although investigating the barge platform concepts rather than the spar buoy concept, introduces model-based control of a floating wind turbine and incorporates the azimuth dependent periodic behavior of the wind turbine into the control design to achieve individual blade pitch actuation and reduce structural loads as a result. The model-based control design in Ref. [5] assumes that all states of the model used by the model-based controller can be measured. In the present work no such assumptions are made and an extended Kalman filter is used to estimate the structural states of the wind turbine as well as blade specific wind speeds and wave forces affecting the wind turbine, using only what is thought to be realistically available sensors.

MPC is typically chosen for its ability to handle constraints both hard and soft and the ability to incorporate knowledge of future disturbances into the prediction horizon of the controller. Performance during extreme events has not been examined and with the conservative tuning of the presented controller no rate limits are in danger of being violated in the present work. During switching between operating regions the constraint handling abilities of MPC has been useful and details about this will be elaborated later in this work. MPC also enables the use of soft constraints. A soft constraint limit is placed on the generator speed, penalizing violations of 10 percent above nominal generator speed. Further soft constraints could be placed on other states or outputs, if needed.

Prepared using weauth.cls [Version: 2010/06/17 v1.00]

Model predictive control (MPC), both in linear [6] , gain scheduled [7] , nonlinear [8, 9] versions have previously been applied on wind turbines. Ref. [6] and Ref. [7] investigate both partial and full load operation, whereas Ref. [8] and Ref. [9] only examine full load operation. The present work considers both partial and full load operation as well smooth switching between the two regions of operation.

Ref. [6] switches between 4 controllers to handle the full wind speed range, all linearized around an equilibrium point for a given wind speed. Ref. [7] enables switching between as many controllers as desired, they are however also linearized around equilibrium points for specific wind speeds and the more the wind speed, rotor speed and blade pitch angles deviate from their equilibrium points the more the linearized dynamical model used by the controller deviates from the real wind turbine dynamics. Linear parameter variable (LPV) control [10] where wind speed as well as pitch angle and rotor speed are used as scheduling variables overcome the problem of nonlinearities dependant of several scheduling variables. For an increasing number of scheduling variables, the offline synthesis of the controller is rendered not piratically implementable. With the added complexity of constraint handling within a prediction horizon the offline synthesis of a controller is close to impossible. Instead, the presented controller relinearize the control design model in each sample and a controller is synthesized online giving a controller designed with the present nonlinearities taken into account. The presented controller is thus an intermediate step between linear and nonlinear MPC.

The current contribution is structured as follows: The model used by the extended Kalman filter and the model predictive controller is presented in the first section. The next section introduces the general theory behind the extended Kalman filter and the model predictive controller. On top of the model predictive control layer is a high-level layer determining which region of operation the controller should be operating. The finer details of this high-level layer is presented in the fourth section. The fifth section presents results where the performance of the presented controller is compared to the performance of benchmark proportional-integral controller. Finally conclusions are drawn.

2. MODEL FOR CONTROLLER DESIGN

In this section the model used by the extended Kalman filter and the model predictive controller is presented. The presented model is also discussed in Ref. [11] where focus in on wind speed and wave force estimation.

2.1. Aerodynamic model

The aerodynamic model used in this work is based on the single-state dynamic inflow model proposed in Ref. [12] with the extension that the effective wind speed of the individual blades rather than a rotor-wide effective wind speed are included. This extension of the model enables individual pitch control in future work.

For the blades $i = 1, \ldots, n_b$, the aerodynamic torque Q_i and thrust force T_i are the integrated aerodynamic forces over the blade span

$$T_i(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_0^R F_n(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{1}$$

$$Q_i(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_0^R rF_t(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{2}$$

where $F_n(\cdot)$ and $F_t(\cdot)$ are local blade forces normal and tangential to rotor-plane. The local tangential induced velocities are assumed quasi-steady and the local axial induced wind speeds $v_{n,i}(r)$ are given by the averaged induced axial wind speed $\bar{v}_{n,i}$ and the quasi-steady distribution of the axial induction factor

$$v_{n,i}(r) = \frac{a_n^{qs}(\lambda_i, \theta_i, r)}{\bar{a}_n^{qs}(\lambda_i, \theta_i)} \bar{v}_{n,i} \tag{3}$$

where

$$\bar{a}_n^{qs}(\lambda_i, \theta_i) = \frac{1}{R} \int_0^R a_n^{qs}(\lambda_i, \theta_i, r)dr \tag{4}$$

The temporal dynamics of the averaged axial induced velocity is governed by a first order ordinary differential equation

$$\bar{v}_{n,i} = \frac{1}{\tau_i s + 1} V_{rel,i} \bar{a}_n^{qs}(\lambda_i, \theta_i) \tag{5}$$

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

with the time constant

$$\tau_i = \frac{1}{2}\frac{1.1R}{V_{rel,i} - 1.3\bar{v}_{n,i}} \tag{6}$$

based on the same assumption as in Ref. [12] .

The aerodynamic torque and thrust of each blade comprises the rotor-wide entities

$$Q = \sum_i^{n_b} Q_i \text{ and } T = \sum_i^{n_b} T_i \tag{7}$$

Strain gauges (SG) are placed at root of the blade at the radial distance r_{SG} and projected into the normal and tangential directions of the rotor plane. If the blades where assumed rigid and massless the aerodynamic bending moments

$$Q_{n,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_{r_{SG}}^{R} rF_n(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{8}$$

$$Q_{t,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) = \int_{r_{SG}}^{R} rF_t(V_{rel,i}, \Omega, \theta_i, v_{n,i}(r), r)dr \tag{9}$$

could directly be obtained from the strain gauges and used to estimate blade specific wind speeds. However, the measured out-of-plane root bending moment $Q_{n,i}^{SG}$ is affected by the centrifugal loading of the blades and is thus lower than the solely aerodynamic out-of-plane root bending moment. Several methods exist to make the out-of-plane sensor usable in the estimation algorithm: The model can be extended to include the structural properties of the blade or the estimation algorithm can assign smaller importance of the sensor by increasing the expected output variance of the sensor used to tune the estimation algorithm. The measured in-rotor-plane root bending moment $Q_{t,i}^{SG}$ is heavily influenced by the gravitational loading of the blades varying with the azimuth angle of the blades. The aerodynamic moments are approximated by the blade root bending moments in the following way

$$Q_{n,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) \approx Q_{n,i}^{SG} + \nu_{qn,i}, \qquad \nu_{qn,i} \in N(0, \sigma_{qn}^2) \tag{10}$$

$$Q_{t,i}(V_{rel,i}, \Omega, \theta_i, \bar{v}_{n,i}) \approx Q_{t,i}^{SG} - F_{g,blade}\sin\phi_{b,i} + \nu_{qt,i}, \qquad \nu_{tn,i} \in N(0, \sigma_{qt}^2) \tag{11}$$

where $F_{g,blade}$ is the gravitational loading on the individual blades and $\phi_{b,i}$ is the azimuth angle of the individual blades and the zero-mean Gaussian distributed noise contributions $\nu_{qn,i}$ and $\nu_{qt,i}$ represent the uncertainties caused by blade dynamics, blade deformation, centrifugal stiffening etc.

2.2. Wind spectra

For rotor-wide wind speed spatially averaged models, based on e.g. the turbulence model of Ref. [13] , are suitable. But for the individual wind speeds of each blade, the strongest source of wind speed variation is the periodic contribution caused by wind shear, tower shadow and the spatial structure of the turbulent field [14]. As a result the wind variation is modeled as a periodically varying wind speed depending on the rotation speed of rotor. The total wind speed for blade i is the sum of a rotor-wide mean wind speed V_m and a blade specific turbulent wind speed $V_{t,i}$

$$V_i = V_m + V_{t,i} \tag{12}$$

where the turbulent wind speed spectrum is given by the rotational speed of the rotor

$$V_{t,i} = \frac{\omega_{n,v}^2}{s^2 + 2\omega_{n,v}\zeta_{v,i}s + \omega_{n,v}^2}\xi_{v,i}, \quad \xi_{v,i} \in N(0, \sigma_v^2) \tag{13}$$

The damped frequency of the turbulent wind field is equal to the rotational speed of the rotor $\omega_{d,v} = \Omega$. Giving a natural frequency dependent of damped frequency and damping ratio $\omega_{n,v} = \omega_{d,v}/\sqrt{1-\zeta_v^2}$. For a constant periodic wind speed the damping ratio should ideally be zero, numerical conditioning especially with regards to the synthesis of model-based controllers, e.g. LQ, is however eased by adding a little damping in the model and $\zeta_v = 0.01$ is chosen. The rotor-wide mean wind speed is modeled as a parameter rather than a state and a first order filter driven by the mean of the individual wind speed estimates of the current sample determines the mean wind speed in the next sample

$$V_m = \frac{1}{\tau s + 1}V, \quad V = \frac{1}{n_b}\sum_i^{n_b} V_i \tag{14}$$

where a time constant $\tau = 1/(\Omega \text{ rad/s})$ has shown good performance with regards to the estimation of the blade specific wind speeds.

2.3. Hydrodynamic forces

The offshore tower either in a floating or fixed configuration is subjected to wave forces. If these forces are not taken into account, the tower top displacement will be ascribed solely to the aerodynamic thrust resulting in a erroneous estimation of wind speed. The hydrodynamic forces on a moving body with velocity v in a fluid with the oscillating flow velocity u can be described by the semi-empirical Morrison equation [1] .

$$F_h = \rho_w V_b \dot{u} + \rho(C_m - 1)V_b(\dot{u} - \dot{v}) + \frac{1}{2}\rho_w C_d A_b(u - v)|u - v| \tag{15}$$

where C_m and C_d are the inertia and drag coefficients of the floating body, ρ_w is the mass density of the fluid and V_b and A_b are volume and area of the body submerged in the fluid. The Morrison equation can be simplified to

$$F_h = \kappa_1 \dot{u} + \kappa_2 \dot{v} + \kappa_3(u - v)|u - v| \tag{16}$$

where the constants κ_1,κ_2 and κ_3 contain the aforementioned coefficients, volume etc. The third term describing the drag force, is assumed to be negligible, leading to $\kappa_3 = 0$. The first term is rewritten to a wave force $F_w = \kappa_1 \dot{u}$. The modeled spectrum of the stochastic oscillating wave force is fitted to the measured spectrum of the water acceleration

$$F_w = \frac{k\omega_{n,w}^2}{s^2 + 2\omega_{n,w}\zeta_w s + \omega_{n,w}^2}\xi_w, \quad \xi_w \in N(0, \sigma_w^2) \tag{17}$$

giving a damped frequency $\omega_{d,w} = \frac{2\pi}{T_p}$ defined by the peak spectral period T_p. In the work of Ref. [15] values for the peak spectral period and significant wave height are given as $T_p = 10$ s and $H_s = 6$ m and crude fitting of the water acceleration power spectral density function to the model (17) gives a damping ratio $\zeta_w = 0.125$ and a steady state gain k of 0.2. The natural frequency can then be determined as $\omega_{n,w} = \omega_{d,w}/\sqrt{1 - \zeta_w^2}$. The stochastic input w_w is assumed to be zero-mean Gaussian distributed white noise. Simulations (not presented here) have shown that the wave force estimation is not sensitive to whether or not the correct peak spectral period is known, correct information about the peak spectral period is of higher importance when used in a model-based control algorithm. The correct peak spectral period can be estimated online via the estimated wave force. The peak of the power spectral density can by computed using various algorithms e.g. Fast Fourier Transform (FFT) algorithms. The peak spectral period can then be updated, e.g. through a first order dynamic filter with a time constant in the range of tens of minutes.

2.4. Tower and spar buoy

The fore-aft motion of the wind turbine can be comprised of three degrees of freedom: Tower bending and pitch and surge of the entire wind turbine. Pitch describes the angle which the turbine is rotated with regards to the vertical axis and surge indicates the displacement of the wind turbine in the wind direction, see also Fig. 1 for a depiction of the mentioned degrees of freedom. In this work, the wind turbine pitch (*tp*) angle is defined as the pitch angle of the wind turbine at the bottom, denoted α. For a rigid tower and spar buoy, the wind turbine pitch angle is related to the relative displacement of the top and bottom of the wind turbine by a simple trigonometric relation

$$y_{tp} = h \sin \alpha \tag{18}$$

where h is the height of the wind turbine. For small variations of α, the pitch displacement be approximated as

$$y_{tp} \approx h\alpha \tag{19}$$

In this work, the surge displacement y_{surge} is defined as the displacement of the bottom of the floating wind turbine relative to point where the turbine would be if no wind or waves where affecting it. The relative displacement of the tower top, where turbine pitch y_{tp} and tower bending (*tb*) y_{tb} are included, is denoted y_{rel} and is given as

$$y_{rel} = y_{top} - y_{surge} = y_{tp} + y_{tb} \tag{20}$$

The tower bending and pitch of the floating wind turbine can be described by two interconnected mass-damper-spring systems (M_{tp}, D_{tp}, K_{tp}) and (M_{tb}, D_{tb}, K_{tb}) where the first system models the pitch motion of the wind turbine and the second system models the bending of the tower. The top of the wind turbine is affected by the aerodynamic thrust forces of each blade T_i, which are functions of the relative wind speed seen from the wind turbine top $V_{rel,i} = V_i - \dot{y}_{rel}$. The bottom of the wind turbine is affected by the hydrodynamic forces (16) consisting of a wave force and a turbine acceleration term. A turbine acceleration term $\kappa_2 \ddot{y}_\alpha$ can be added to the inertial mass of the turbine pitch resulting in a

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

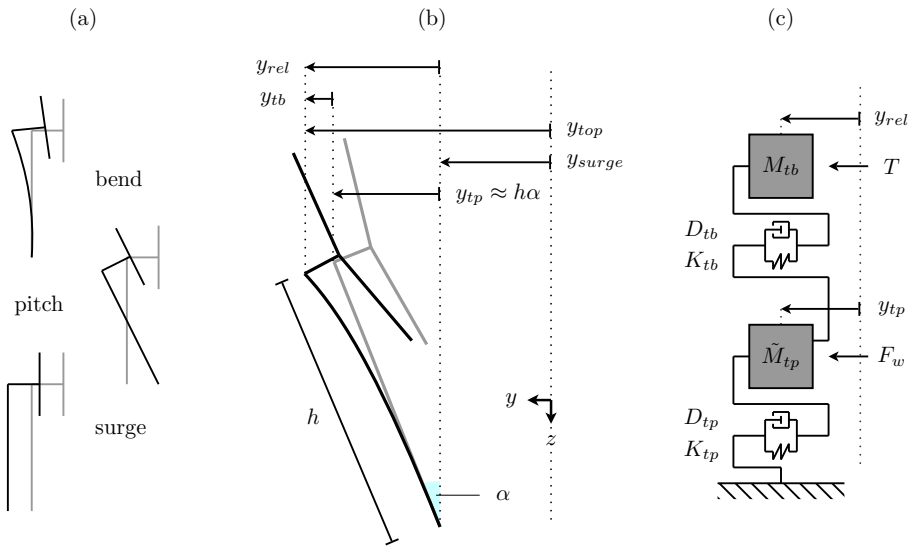


Figure 1. Degrees of freedom of tower in fore-aft direction. (a) Degrees of freedom: tower bend, pitch and surge, only tower bend and pitch are included in the model. (b) The relative position y_{rel} is the difference between the position of the tower top y_{top} and the floater bottom y_{surge}. The sum of turbine pitch displacement y_{tp} and tower bending displacement y_{tb} equals y_{rel}. (c) The floating tower modeled as two interconnected mass-damper-spring systems.

new mass $\tilde{M}_{tp} = M_{tp} + \kappa_2$ and the resulting system

$$M_{tb}\ddot{y}_{rel} + D_{tb}\dot{y}_{tb} + K_{tb}y_{tb} = T \tag{21a}$$

$$\tilde{M}_{tp}\ddot{y}_{tp} + D_{tp}\dot{y}_{tp} + K_{tp}y_{tp} = F_w + D_{tb}\dot{y}_{tb} + K_{tb}y_{tb} \tag{21b}$$

describes the fore-aft degrees of freedom of the floating wind turbine, see Fig. 1 for a depiction of the two interconnected mass-damper-spring systems. It is expected that an acceleration sensor in the tower top and an angle measurement in the bottom are realistic assumptions of available sensors. The velocity and acceleration of the bottom angles can then be calculated via numerical difference.

2.5. Drive-train

The drive-train connects the rotor to the generator through a low speed shaft, a gearbox and a high speed shaft. The drive-train flexibility is a modeled in the low speed shaft coordinate system

$$I_r\ddot{\phi}_r + D_s\dot{\phi}_\Delta + K_s\phi_\Delta = Q \tag{22}$$

$$I_g N_g^2 \frac{\ddot{\phi}_g}{N_g} - D_s\dot{\phi}_\Delta - K_s\phi_\Delta = -Q_g N_g \tag{23}$$

where N_g is the gear ratio, I_r and I_g are the moments of inertia of the rotor and generator, K_s and D_s are the spring and damping constants. It should also be mentioned that the following definitions are introduced: $\dot{\phi}_r \equiv \Omega$ is rotor speed, $\dot{\phi}_g \equiv \Omega_g$ is generator speed and $\phi_\Delta \equiv \phi_r - \frac{\phi_g}{N_g}$ is the angular torsion of the drive-shaft in the low speed shaft coordinate system.

2.6. Actuators

The actuators are assumed linear under the assumption that a low level controller, e.g. PID or some type of nonlinear controller e.g. [16], is operating in closed loop with the actuator mechanics. The closed loop actuator is described with 2nd order dynamics, an approximation which under the proper conditions can be justified [17]

$$\ddot{\theta}_i + 2\zeta_\theta\omega_\theta\dot{\theta}_i + \omega_\theta^2\theta_i = \omega_\theta^2\theta_{i,ref} \tag{24a}$$

subject to

$$\begin{bmatrix}\theta_{min}\\ \dot{\theta}_{min}\end{bmatrix} \leq \begin{bmatrix}\theta_i\\ \dot{\theta}_i\end{bmatrix} \leq \begin{bmatrix}\theta_{max}\\ \dot{\theta}_{max}\end{bmatrix} \tag{24b}$$

where ω_θ and ζ_θ are the natural frequency and damping ratio of the actuator and θ_{ref} is the reference signal from the controller. The generator torque actuator is assumed to be described with 1st order dynamics [18]

$$\dot{Q}_g + \tau_g^{-1} Q_g = \tau_g^{-1} Q_{g_{ref}} \tag{25a}$$

subject to

$$\begin{bmatrix}Q_{g,min}\\ \dot{Q}_{g,min}\end{bmatrix} \leq \begin{bmatrix}Q_g\\ \dot{Q}_g\end{bmatrix} \leq \begin{bmatrix}Q_{g,max}\\ \dot{Q}_{g,max}\end{bmatrix} \tag{25b}$$

where τ_g is the time constant of the generator torque actuator and $Q_{g_{ref}}$ is the reference signal from the controller.

2.7. Combined model

The ordinary differential equations of the submodels, are gathered in a state space ordinary differential equation function and time-discretized to obtain the state progress equation

$$\boldsymbol{x}_{k+1} = \underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{26a}$$

and the ouputs are gathered in an output state space function

$$\boldsymbol{y}_k = \boldsymbol{g}_{\boldsymbol{y}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{26b}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the input vector and $\boldsymbol{y}$ is the measurement vector. The vectors are comprised by the following variables

$$\begin{aligned}
\boldsymbol{x}_{\boldsymbol{struct}} &= [\Omega_r\ \Omega_g\ \phi_\Delta\ y_\alpha\ \dot{y}_\alpha\ y_\beta\ \dot{y}_\beta]^T\\
\boldsymbol{x}_{\boldsymbol{aero,wave}} &= [V_{t,i}\ \dot{V}_{t,i}\ \bar{v}_{n,i}\ F_w\ \dot{F}_w]^T\\
\boldsymbol{x}_{\boldsymbol{actuator}} &= [\theta_i\ \dot{\theta}_i\ Q_g]^T\\
\boldsymbol{x} &= [\boldsymbol{x}_{\boldsymbol{struct}}\ \boldsymbol{x}_{\boldsymbol{aero,wave}}\ \boldsymbol{x}_{\boldsymbol{actuator}}]^T\\
\boldsymbol{u} &= [\theta_{ref,i}\ Q_{g,ref}]^T\\
\boldsymbol{w} &= [e]^T\\
\boldsymbol{y}_{\boldsymbol{struct}} &= [\Omega_r\ \Omega_g\ \phi_\Delta\ \alpha\ \dot{\alpha}\ \ddot{\alpha}\ \ddot{y}_{top}\ Q_{n,i}^{SG}\ Q_{t,i}^{SG}]^T\\
\boldsymbol{y}_{\boldsymbol{actuator}} &= [\theta_i\ \dot{\theta}_i\ \ddot{\theta}_i\ Q_g\ \dot{Q}_g\ P_e]^T\\
\boldsymbol{y} &= [\boldsymbol{y}_{\boldsymbol{struct}}\ \boldsymbol{y}_{\boldsymbol{actuator}}]^T
\end{aligned}$$

3. MODEL PREDICTIVE CONTROL AND STATE ESTIMATION

The controller presented in this work is based on full state information, which is not available, so an extended Kalman filter is used to estimate the states. The control design model is augmented with disturbance states to achieve offset-free control at steady state, which will be explained in section 3.1. The model predictive controller with and without constraint handling capabilities is presented in section 3.2.

3.1. Extended Kalman filter and augmented disturbance model

The method presented to achieve offset-free reference tracking is comparable to that of a proportional-integral (PI) controller. Instead of integrating the error between reference and measured output as done by the PI controller, the implemented method integrates the error between estimated output and measured output. Discrepancies between model and plant outputs are used to compensate for unmodeled effects and thus obtain offset-free reference tracking by taking the disturbances into account in the controller.

DOI: 10.1002/we
Prepared using weauth.cls

To ensure off-set free performance, the control design model (26) can be augmented with a disturbance model [19, 20], which is used to compensate for actual disturbances and for any plant/model mismatch

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{p} \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} \underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{E}\boldsymbol{d}_k \\ \boldsymbol{d}_k \\ \boldsymbol{p}_k \end{bmatrix}}_{\hat{\underline{\boldsymbol{f}}}(\boldsymbol{x}_k, \boldsymbol{d}_k, \boldsymbol{p}_k, \boldsymbol{u}_k)} + \begin{bmatrix} \boldsymbol{w}_{\boldsymbol{x}} \\ \boldsymbol{w}_{\boldsymbol{d}} \\ \boldsymbol{w}_{\boldsymbol{p}} \end{bmatrix}_k \tag{27a}$$

$$\boldsymbol{y}_k = \underbrace{\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{F}\boldsymbol{p}_k}_{\hat{\boldsymbol{g}}(\boldsymbol{x}_k, \boldsymbol{d}_k, \boldsymbol{p}_k, \boldsymbol{u}_k)} + \boldsymbol{v}_k \tag{27b}$$

where

$$\begin{bmatrix} \boldsymbol{w}_{\boldsymbol{x}} \\ \boldsymbol{w}_{\boldsymbol{d}} \\ \boldsymbol{w}_{\boldsymbol{p}} \end{bmatrix} \in N(\mathbf{0}, \mathbf{R}_{\mathbf{xdp}}), \text{ and } \boldsymbol{v} \in N(\mathbf{0}, \mathbf{R}_{\mathbf{y}})$$

where $\boldsymbol{d} \in \mathbb{R}^{n_d}$ are state disturbances and $\boldsymbol{p} \in \mathbb{R}^{n_p}$ are output disturbances and where $\mathbf{R}_{\mathbf{xdp}} \in \mathbb{R}^{(n_x+n_d+n_p)\times(n_x+n_d+n_p)}$ is variance of the states and disturbances and $\mathbf{R}_{\mathbf{y}} \in \mathbb{R}^{n_y \times n_y}$ is the variance of the output.

The disturbance model $(\mathbf{E}, \mathbf{F})$ can be structured in many ways but from a practical perspective is should be simple to create. As prescribed by Ref. [20] there are as many disturbances as there measured outputs. If a state is measured directly (e.g. generator speed), then the corresponding disturbance is assigned be an input disturbance. If the measurement is a combination of states (e.g. generator power, which is the product of the states generator speed and generator torque), then the corresponding disturbance is assigned to be an output disturbance.

Because the disturbances cannot be measured they have to be estimated. A discrete time predictive Kalman filter is designed to estimate states and disturbances. The predictive Kalman filter, which only bases its estimations on the previous sample, is chosen instead of the ordinary Kalman, which bases its estimations on the previous and current sample. The choice of the predictive Kalman filter is necessary since some measurements include a direct term from input to output, e.g. pitch acceleration and generator torque rate, and the problem of causality would arise if an ordinary Kalman filter was implemented. Furthermore, the predictive Kalman filter gives the computationally expensive MPC algorithm an entire sample time period to compute the control signal.

The estimated states and disturbances are denoted $[\hat{\boldsymbol{x}}\ \hat{\boldsymbol{d}}\ \hat{\boldsymbol{p}}]^T_{k|k-1}$, where $k|k-1$ means the estimation at time k given by the information at time $k-1$. The *a posteriori* estimation of the states is given by

$$\begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{d}} \\ \hat{\boldsymbol{p}} \end{bmatrix}_{k|k} = \begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{d}} \\ \hat{\boldsymbol{p}} \end{bmatrix}_{k|k-1} + \mathbf{L}_k[\boldsymbol{y}_k - \hat{\boldsymbol{g}}(\hat{\boldsymbol{x}}_{k|k-1}, \hat{\boldsymbol{d}}_{k|k-1}, \hat{\boldsymbol{p}}_{k|k-1}, \boldsymbol{u}_k)] \tag{28}$$

Enabling a *a priori* estimation of the one-step-ahead prediction of the states

$$\begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{d}} \\ \hat{\boldsymbol{p}} \end{bmatrix}_{k+1|k} = \hat{\underline{\boldsymbol{f}}}(\hat{\boldsymbol{x}}_{k|k}, \hat{\boldsymbol{d}}_{k|k}, \hat{\boldsymbol{p}}_{k|k}, \boldsymbol{u}_k) \tag{29}$$

where the Kalman gain $\mathbf{L}_k \in \mathbb{R}^{(n_x+n_d+n_p)\times n_y}$ and output error covariance $\boldsymbol{\Psi}_k$

$$\mathbf{L}_k = \mathbf{P}_{k|k-1}\hat{\mathbf{C}}^T_{k|k-1}\boldsymbol{\Psi}_k^{-1} \tag{30a}$$

$$\boldsymbol{\Psi}_k = \hat{\mathbf{C}}_{k|k-1}\mathbf{P}_{k|k-1}\hat{\mathbf{C}}^T_{k|k-1} + \mathbf{R}_{\mathbf{y}} \tag{30b}$$

is updated by the discrete time recursive Riccati equation

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k\hat{\mathbf{C}}_{k|k-1}\mathbf{P}_{k|k-1} \tag{30c}$$

$$\mathbf{P}_{k+1|k} = \hat{\mathbf{A}}_{k|k}\mathbf{P}_{k|k}\hat{\mathbf{A}}^T_{k|k} + \mathbf{R}_{\mathbf{xdp}} \tag{30d}$$

where $\hat{\mathbf{A}} = [\nabla_{\boldsymbol{x}}\hat{\underline{\boldsymbol{f}}}\ \nabla_{\boldsymbol{d}}\hat{\underline{\boldsymbol{f}}}\ \nabla_{\boldsymbol{p}}\hat{\underline{\boldsymbol{f}}}]$ and $\hat{\mathbf{C}} = [\nabla_{\boldsymbol{x}}\hat{\boldsymbol{g}}\ \nabla_{\boldsymbol{d}}\hat{\boldsymbol{g}}\ \nabla_{\boldsymbol{p}}\hat{\boldsymbol{g}}]$, which are calculated at the time given by the time index k.

3.2. Relinearized model predictive control

In this section, the estimated disturbances $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{p}}$ are assumed to be constant and are used as parameters rather than dynamic states. Hence, the disturbances are not functional arguments of the various equations but implied constant contributions embedded in the equations.

The relinearized model predictive controller (RLMPC) entails the computation of the control signal within a prediction horizon in the range $k = (0, \ldots, N-1)$. The RLMPC is formulated as a dual mode horizon where the first part, i.e $k = (0, \ldots, N-1)$, is considered constrained. In the second horizon, i.e. $i = (N, \ldots, \infty)$, it is assumed that the plant has reached a state where the unconstrained solution is feasible [21].

The dual mode optimization problem is

$$\min \sum_{k=0}^{N-1} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \sum_{k=0}^{N-1} \frac{1}{2} \boldsymbol{\sigma}_k^T \mathbf{W}_\sigma \boldsymbol{\sigma}_k + \sum_{k=N}^{\infty} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{31a}$$

where the stagewise cost function

$$\phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \frac{1}{2} \boldsymbol{g_z}(\boldsymbol{x}_k, \boldsymbol{u}_k)^T \mathbf{W_z} \boldsymbol{g_z}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \frac{1}{2} (\boldsymbol{r} - \boldsymbol{g_r}(\boldsymbol{x}_k, \boldsymbol{u}_k))^T \mathbf{W_r} (\boldsymbol{r} - \boldsymbol{g_r}(\boldsymbol{x}_k, \boldsymbol{u}_k))$$

consist of two terms: The first term seeks to minimize dynamic variations given by $\boldsymbol{g_z}(\cdot)$ such as e.g. velocities, accelerations. The second term seeks to drive the plant reference outputs $\boldsymbol{g_r}(\cdot)$ towards the desired reference $\boldsymbol{r}$, e.g. generator power and generator speed. An additional cost term, only included in the first part of the prediction horizon, seeks to minimize the violation of the soft constraints $\boldsymbol{\sigma}$. The optimization problem is subject to an initial constraint

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}} \tag{31b}$$

where $\bar{\boldsymbol{x}}$ is the current state estimate. Throughout the entire prediction horizon $(k = 0, \ldots, \infty)$, the optimization problem is subject to the state progress equation constraint (26a). Whereas the soft and hard inequality constraints are only active in the first part of the prediction horizon $(k = 0, \ldots, N-1)$

$$\boldsymbol{g_s}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{\sigma}_k \leq \boldsymbol{s} \tag{31c}$$

$$\boldsymbol{g_h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{h} \tag{31d}$$

The nonlinear equations are assumed linear throughout the entire prediction horizon in order to ease the problem solving, as the nonlinear programming problem reduces to a quadratic programming problem, which can be solved using either interior-point or active-set algorithms [22]. The equations for the reference outputs $\boldsymbol{g_r}(\cdot)$, the dynamic outputs $\boldsymbol{g_z}(\cdot)$ and the soft $\boldsymbol{g_s}(\cdot)$ and hard $\boldsymbol{g_h}(\cdot)$ constraint equations are linearized around $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$, where $\bar{\boldsymbol{u}}$ can be chosen arbitrarily since all the equations are affine in the input, meaning that e.g. $\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ can be reformulated to $\boldsymbol{a}(\boldsymbol{x}_k) + \boldsymbol{b}(\boldsymbol{x}_k)\boldsymbol{u}_k$ etc. The state progress equation can be approximated by the linearized function

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \underline{\mathbf{A}} \boldsymbol{x}_k + \underline{\mathbf{B}} \boldsymbol{u}_k + \boldsymbol{\delta} \tag{32}$$

where the constant contribution is

$$\boldsymbol{\delta} = -\underline{\mathbf{A}} \bar{\boldsymbol{x}} - \underline{\mathbf{B}} \bar{\boldsymbol{u}} + \underline{\boldsymbol{f}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$$

the reference tracking function $\boldsymbol{g_r}(\cdot)$ can be approximated by

$$\boldsymbol{g_r}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{C_r} \boldsymbol{x}_k + \mathbf{D_r} \boldsymbol{u}_k + \boldsymbol{\gamma_r} \tag{33}$$

where

$$\boldsymbol{\gamma_r} = -\mathbf{C_r} \bar{\boldsymbol{x}} - \mathbf{D_r} \bar{\boldsymbol{u}} + \boldsymbol{g_r}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) \tag{34}$$

The other output functions $\boldsymbol{g_z}(\cdot)$, $\boldsymbol{g_s}(\cdot)$ and $\boldsymbol{g_h}(\cdot)$ can be linearized in a similar manner. The linearized stagewise cost function can be put on a more general form

$$\phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \frac{1}{2} (\boldsymbol{x}_k^T \mathbf{Q} \boldsymbol{x}_k + \boldsymbol{u}_k^T \mathbf{R} \boldsymbol{u}_k + 2\boldsymbol{x}_k^T \mathbf{M} \boldsymbol{u}_k + 2\boldsymbol{q}^T \boldsymbol{x}_k + 2\boldsymbol{r}^T \boldsymbol{u}_k) \tag{35}$$

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

where

$$\mathbf{Q} = {\mathbf{C_r}}^T \mathbf{W_r} \mathbf{C_r} + {\mathbf{C_z}}^T \mathbf{W_z} \mathbf{C_z} \tag{36a}$$
$$\mathbf{R} = {\mathbf{D_r}}^T \mathbf{W_r} \mathbf{D_r} + {\mathbf{D_z}}^T \mathbf{W_z} \mathbf{D_z} \tag{36b}$$
$$\mathbf{M} = {\mathbf{C_r}}^T \mathbf{W_r} \mathbf{D_r} + {\mathbf{C_z}}^T \mathbf{W_z} \mathbf{D_z} \tag{36c}$$
$$\boldsymbol{q}^T = [\boldsymbol{\gamma_r} - \boldsymbol{r}]^T \mathbf{W_r} \mathbf{C_r} + {\boldsymbol{\gamma_z}}^T \mathbf{W_z} \mathbf{C_z} \tag{36d}$$
$$\boldsymbol{r}^T = [\boldsymbol{\gamma_r} - \boldsymbol{r}]^T \mathbf{W_r} \mathbf{D_r} + {\boldsymbol{\gamma_z}}^T \mathbf{W_z} \mathbf{D_z} \tag{36e}$$

The second part of the optimization problem can be reduced to a terminal cost, consisting of a quadratic $\boldsymbol{\Pi}$ and a linear cost term $\boldsymbol{\pi}$

$$\sum_{k=N}^{\infty} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_N^T \boldsymbol{\Pi} \boldsymbol{x}_N + \boldsymbol{\pi}^T \boldsymbol{x}_N \tag{37}$$

The quadratic cost for the terminal cost is found by the discrete-time algebraic Riccati equation (DARE)

$$\boldsymbol{\Pi} = \mathbf{Q} + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \underline{\mathbf{A}} - \tilde{\mathbf{M}} \mathbf{K} \tag{38}$$

where to simplify notation, the matrices

$$\tilde{\mathbf{M}} = \mathbf{M} + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \underline{\mathbf{B}} \text{ and } \tilde{\mathbf{R}} = \mathbf{R} + \underline{\mathbf{B}}^T \boldsymbol{\Pi} \underline{\mathbf{B}}$$

are introduced, leading to the feedback gain $\mathbf{K} = \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{M}}^T$. The linear terminal cost term $\boldsymbol{\pi}$ is determined from the same recursion as the discrete-time algebraic Riccati equation

$$\boldsymbol{\pi} = \boldsymbol{q} - \tilde{\mathbf{M}} \boldsymbol{\kappa} + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \boldsymbol{\delta} + \underline{\mathbf{A}}^T \boldsymbol{\pi} \tag{39}$$

where the control action contribution $\boldsymbol{\kappa}$

$$\boldsymbol{\kappa} = \tilde{\mathbf{R}}^{-1} [\boldsymbol{r} + \underline{\mathbf{B}}^T \boldsymbol{\Pi} \boldsymbol{\delta} + \underline{\mathbf{B}}^T \boldsymbol{\pi}] \tag{40}$$

is part of the optimal control law

$$\boldsymbol{u}_k = -\mathbf{K} \boldsymbol{x}_k - \boldsymbol{\kappa} \tag{41}$$

which is assumed to be active during the second part of the optimization problem. If no inequality constraints are present, the presented control law can be used as the controller of the plant. The DARE should be solved using a specialized DARE solver [23] to achieve fast and robust results and the linear cost term can be found from (39)

$$\boldsymbol{\pi} = [\mathbf{I} - \underline{\mathbf{A}}^T + \mathbf{K}^T \underline{\mathbf{B}}^T]^{-1} [\boldsymbol{q} - \mathbf{K}^T [\boldsymbol{r} + \underline{\mathbf{B}}^T \boldsymbol{\Pi} \boldsymbol{\delta}] + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \boldsymbol{\delta}] \tag{42}$$

Further information about the presented RLMPC controller can be found in [24].

4. HYBRID CONTROLLER SETUP

The controller switches between two regions of operation, thereby between two control objectives, namely full and partial load. A switching mechanism determines which control objective is presently active.

The primary static objective of the wind turbine control system is to optimize power production for the given wind speed. Load reduction is a typical secondary objective, which shall however not be discussed further in this section. The primary objective for a given wind speed can be formulated as the constrained minimization of nonlinear quadratic cost function concerning generator power. An additional term concerning generator speed is included in the cost function, this term is only active when the first term concerning power is zero. The steady state constrained optimization problem for a given wind speed is

$$\min_{\Omega_g, \theta} (P_e - P_{nom})^2 + w(\Omega_g - \Omega_{g,nom})^2, \tag{43a}$$

where

$$w = \begin{cases} 0 & \text{for} \quad P_e \neq P_{nom} \\ 1 & \text{for} \quad P_e = P_{nom} \end{cases}$$

subject to

$$\mathbf{0} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{43b}$$
$$\Omega_g \in [\Omega_{g,low}, \Omega_{g,nom}] \tag{43c}$$
$$\theta \in [\theta_{min}, \theta_{max}] \tag{43d}$$

where the equality constraint (43b) ensures steady state operation. The generator speed and pitch angle are limited to a certain ranges (43c) and (43d). There are also other constraints such pitch rate and acceleration etc. but these velocity constraints are not active during the present steady state optimization sweep. Above rated wind speeds, where the generator power is at its nominal value, there is no unique solution for the optimization problem. The pitch angle and generator speed could have any number of solutions as long as they are on the appropriate contour line of the C_P curve. The generator speed weight w ensures that for above rated wind speed there is a unique solution with generator speed at its nominal value.

The optimization gives the characteristic diagrams for generator power, generator speed and pitch angle versus wind speed as seen in Figure 2. Four regions of operation $O \in (I, \ldots, IV)$ also denoted $(O_I, \ldots, O_{IV})$ respectively, are derived from (43). The primary control objective in the partial load regions, $O_I - O_{III}$ is to maximize the amount of power converted from the wind to generator, while keeping within the operating range of the generator speed. If a wind speed measurement or estimate is available, the controller can aim to track the generator speed as a function wind speed according to Figure 2. For above rated wind speeds, O_{IV}, the control objective is to regulate the generator power and speed to their nominal values.

The resulting hybrid controller has a setup as shown in Figure 3. The pitch angle should should ideally be at its optimum in for a given wind speed in O_I-O_{III} as seen in Figure 2. Instead, the collective pitch angle is chosen to be at its optimal value in these regions to simplify the control problem. The choice of fixating the collective pitch angle at lower wind speeds means that only the generator torque is used as a primary control variable in the partial load controller. In this region the collective pitch angle is controlled at the prescribed value. The constant collective pitch angle in O_I to O_{III} has little impact of the aerodynamic efficiency due to the flatness of the C_P curve.

The supervisor block in Figure 3 follows the switching conditions listed in Table I to determine which controller is active. The set of switching conditions depend on whether constant torque or constant power are used by the controller and for a floating turbine where is not possible control as aggressive as on a normal wind turbine, additional switch conditions regarding generator speed have been added to make the overall performance more robust. If the controller is in the partial load regions $O_I - O_{III}$ it will switch to the full load region if the electrical generator power exceeds it nominal value. If

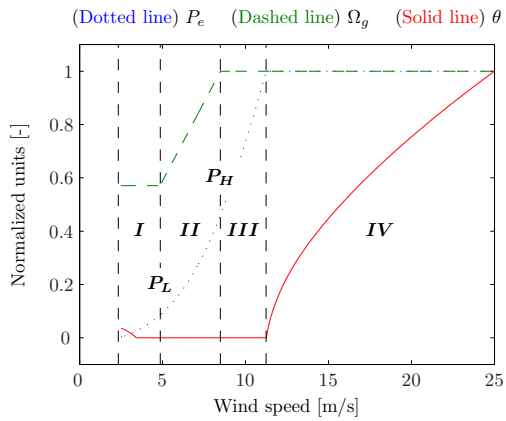


Figure 2. Sweep of wind speeds showing the steady state values of the primary variables of the wind turbine. Notice the pitch angle would normally have slightly different values for wind speeds below rated wind speed. The pitch angle in this case have been constrained and is thus different from its unconstrained value. This is however not expected to have significant influence on the performance of the controller.

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

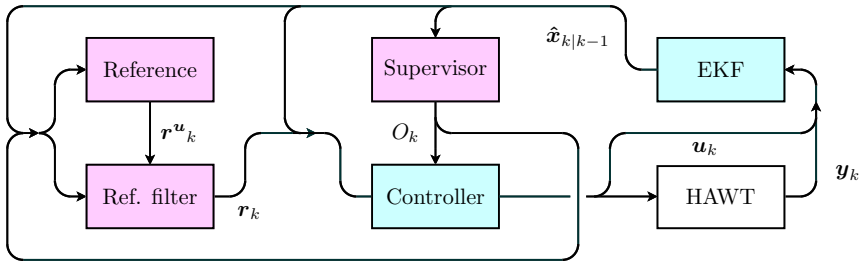


Figure 3. Setup of the hybrid controller. An extended Kalman filter provides estimates of states used by other blocks in the diagram. Supervisor block provides partial or full load control objectives to controller depending on switching conditions. Reference and reference filter blocks provide references for the controller to track depending on whether partial or full load operation is active.

the controller is in the full load region O_{IV} it will switch to the partial load regions if the collective pitch angle is below its optimal value.

One of two modes of full load operation can be chosen, either constant torque or constant power. Constant torque operation typically leads to lower loads at the expense of generator power variability. Constant power operation lowers the generator power variability at the expense of higher loads and also conflicts with the generator speed control objective. The reference tracking objectives for partial and full load operation are

$$\boldsymbol{g}_{\boldsymbol{r},\boldsymbol{partial}}(\cdot) = [\Omega_g \ \theta_{ref,col}]^T \tag{44}$$

$$\boldsymbol{g}_{\boldsymbol{r},\boldsymbol{full}}(\cdot) = [\Omega_g \ P_{ge} \ Q_{g,ref}]^T \tag{45}$$

$$\boldsymbol{r}_{\boldsymbol{partial}} = [\Omega_g(\hat{v}) \ \theta^{opt}]^T \tag{46}$$

$$\boldsymbol{r}_{\boldsymbol{full}} = [\Omega_{g,nom} \ P_{nom} \ Q_{g,nom}]^T \tag{47}$$

$$\mathbf{W}_{\mathbf{r,partial}} = \text{diag}([w_{p,1} \ w_{p,2}]) \tag{48}$$

$$\mathbf{W}_{\mathbf{r,full}} = \text{diag}([w_{f,1} \ w_{f,2} \ w_{f,3}]) \tag{49}$$

where the weight matrix for the full load operation determines whether constant torque or constant power should be the control strategy of choice. With a careful balancing of $w_{f,2}$ and $w_{f,3}$ a compromise between the two full load control strategies could also be obtained. For the floating wind turbine constant torque operation has been chosen to reduce loads.

4.1. Soft constraint

Three soft constraints are used in the present work. The first is a soft constraint on generator speed and becomes active if generator speed exceeds its nominal value with more than 10 percent. The other two constraints are due to switching between regions of operation. With the constant torque control a soft constraint has been placed on generator torque. The controller is penalized for a generator torque above nominal value. A lower constraint has been placed on the collective blade pitch angle at its optimal value.

4.2. Reference filter

To avoid violent transitions between different operating regions, a filter is inserted between the individual controllers and the reference block. At times of switching between partial and full load operation, the reference filter ensures that the reference sent to the controllers is initially the measured controlled output and that the reference sent to the controllers

S	Switching conditions
(partial $\rightarrow$ full)	$(P_e \geq P_{nom})$
(full $\rightarrow$ partial)$_1$	$(P_e < 0.999P_{nom}) \wedge (\theta < \theta_{min} + 0.05)$
(full $\rightarrow$ partial)$_2$	$(\Omega_g \leq 0.8\Omega_{g,nom})$

Table I. Switching conditions of supervisor block in hybrid controller. The chosen set of conditions depends on the overall controller configuration, e.g. for a constant power non-floating wind turbine, the conditions might be different.

approaches the real reference at a rate specified by the reference filter. The reference to track $\boldsymbol{r}$ is damped by a 1st order filter

$$\boldsymbol{r} = \boldsymbol{y}^{\boldsymbol{r}}_{k_s} + \frac{1}{s\tau_r + 1}(\boldsymbol{r}^{\boldsymbol{u}} - \boldsymbol{y}^{\boldsymbol{r}}_{k_s}) \tag{50}$$

where $\boldsymbol{y}^{\boldsymbol{r}}_{k_s}$ is the measured value of the controlled variable at the time of switch and $\boldsymbol{r}^{\boldsymbol{u}}$ is the undamped reference. At the occurrence of a switch of operating region, the states of the filter are reset. In this work the time constant τ_r has been chosen to be 0.1 s.

5. RESULTS

In this section the proposed RLMPC controller design is tested on the NREL 5MW reference wind turbine [1] in the hydro-aero-servo-elastic code HAWC2 [25] and compared to a benchmark PI-based controller proposed by Ref. [1] . The PI controller has been modified to achieve better generator speed control performance. This has been done by adding a quadratic error gain in the control loop, this enables more control action when the generator speed is further away from nominal than when close. The pitch and generator actuators have been implemented as external dll's coupled to HAWC2. They are modeled as first and second order filters (24) and (25), with hard coded constraints on position and rate.

Three simulations have been performed, at mean wind speeds of 8, 12 and 16 m/s, respectively. The different mean wind speeds results in both partial (8 m/s), full (16 m/s) and combined (12 m/s) load simulations. The simulations include a turbulent wind field Ref. [13] with turbulence intensity is 0.14 and a power law wind shear with a coefficient of 0.14 is also added. Linear irregular wave kinematics based on composition of Airy waves with Wheeler stretching, with a significant wave height H_s = 6 m and peak spectral period T_p = 10 s, are used. The chosen parameters for the hydrodynamic loads are based on Ref. [15] .

Two different RLMPC configurations have been tested:

- RLMPC^1 - Control design model does not include dynamic inflow, but assumes quasi-steady aerodynamics.
- RLMPC^2 - Control design model includes dynamic inflow.

Results for the first simulation with a mean wind speed of 8 m/s can be seen in Figures 4 and 5. The wind speed estimate (not shown here) of the RLMPC controllers is a bit higher than the real wind speed and the generator speed is controlled to a higher level than done by the PI controller. The generator power output is however quite close for all the controllers. The tower pitch cause the generator speed to vary which in turn cause the generator torque controlled by the PI controller to vary. This tower pitch dependant generator torque variation is not seen in RLMPC's as generator speed is tracked as function of estimated wind speed rather than estimated relative wind speed.

The second simulation with a mean wind speed of 12 m/s can be seen in Figures 6 and 7. It can be seen that both the PI controller and the RLMPC's manage to switch between regions of operation. The PI controller however, has some significant generator torque rate changes when switching from full to partial load operation. This sudden change in generator torque is not seen in RLMPC's as the use of a reference filter smoothens switching.

Performance for the third simulation with a mean wind speed of 16 m/s can be seen in Figures 8 and 9. The performance of the PI controller and RLMPC's is quite similar.

Five different sensors are used for comparison of fatigue loads. Fatigue load calculations are based on the standard defined by Ref. [26] . Two different material numbers are used $m = 3$ for steel and $m = 12$ for fiberglass.

- Tower base fore-aft ($m = 3$)
- Tower side-side ($m = 3$)
- Blade root flap-wise ($m = 12$)
- Low speed drive shaft ($m = 3$)
- Yaw drive ($m = 3$)

Another measure of control performance is to compare the root-mean-square (RMS) of selected signals.

- Blade pitch rate ($\sum_i^{n_b}$ RMS$(\dot{\theta}_i - 0)$)
- Generator speed error (RMS$(\Omega_g - \Omega_{g,nom})$)
- Generator power error (RMS$(P_e - P_{nom})$)

Tables II, III and IV present the performance comparison measures normalized with regards to the benchmark PI controller for the three simulations, respectively. A positive number in the tables indicates a percent-wise improvement compared to the PI controller.

Table II compares load reduction performance of the two RLMPC controllers relative to the benchmark PI controller for the first simulation at partial load operation. Changes in tower base fore-aft fatigue load is less than 1 percent both

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

Controller	Tow. base fore-aft	Tow. base side-side	Bl. root flap-wise	Low sp. drive shaft	Yaw drive	Bl. pitch activity	RMS of G. sp. err.	RMS of G. p. err.
RLMPC[1]	0.86	46.44	-19.67	11.02	-9.15	0.00	32.41	0.18
RLMPC[2]	0.67	43.82	-15.49	10.70	-8.35	0.00	31.07	0.17

Table II. Mean wind speed of 8 m/s: Fatigue load reduction etc. compared to PI controller in percent. Positive number means improvement compared to PI controller.

Controller	Tow. base fore-aft	Tow. base side-side	Bl. root flap-wise	Low sp. drive shaft	Yaw drive	Bl. pitch activity	Gen. speed error	Gen. pow. error
RLMPC[1]	9.66	-7.50	21.21	1.99	-1.48	63.89	19.24	4.87
RLMPC[2]	10.79	7.72	18.49	1.49	-1.00	64.79	24.78	6.04

Table III. Mean wind speed of 12 m/s: Fatigue load reduction etc. compared to PI controller in percent. Positive number means improvement compared to PI controller.

Controller	Tow. base fore-aft	Tow. base side-side	Bl. root flap-wise	Low sp. drive shaft	Yaw drive	Bl. pitch activity	Gen. speed error	Gen. pow. error
RLMPC[1]	2.61	-32.33	9.05	3.47	0.61	36.27	14.03	14.03
RLMPC[2]	2.26	-32.95	9.10	3.00	0.69	36.09	16.06	16.06

Table IV. Mean wind speed of 16 m/s: Fatigue load reduction etc. compared to PI controller in percent. Positive number means improvement compared to PI controller.

RLMPC's and no real improvement can be claimed. This is not surprising as collective blade pitch is kept at the optimal value and can not be used to dampen tower pitch with the present tuning of the RLMPC's. The RLPMC's show superior performance when it comes to tower base side-side and low speed drive shaft fatigue load reductions. This is probably caused by the fact that the generator torque of RLMPC's depends on wind speed rather than relative wind speed, giving less variation of the generator torque for the RLMPC's. Blade root flap-wise and yaw drive loads are increased. This could, aside from wrong tuning of the RLMPC's, be caused by a control design model which does not describe the needed degrees of freedom.

Load reduction performance of the two RLMPC controllers relative to the benchmark PI controller for the second simulation with both partial and full load operation is compared in Table III. Tower base fore-aft load reduction is achieved with both of the RLMPC's as they now are able to change the collective blade pitch during full load operation. Blade root flap-wise loads are also reduced, possibly an indirect consequence of reduced tower base fore-aft load reductions. In this simulation the two RLMPC show the biggest difference in performance as RLMPC^2 which includes dynamic inflow in the control design model shows a reduction in tower side-side loads, whereas RLMPC^1 shows an increase in tower side-side loads compared the PI controller. Further simulations, investigating the importance of dynamic inflow in the control design model are needed before any real conclusion can be made in that respect.

In Table IV load reduction performance of the three RLMPC controllers during full load operation is presented. The trends of the results are similar to those of the second simulation. Both RLMPCs show superior performance compared to the PI controller for fatigue load reduction, pitch activity reduction and generator speed reference tracking. The only load increase is on tower side-side, an degree of freedom not included in the control design model of the RLPMCs and hence performance improvement cannot be expected.

6. CONCLUSION

In the presented work a floating wind turbine has been controlled with collective blade pitch. The presented framework enables individual blade pitch for all wind speeds, but preliminary implementations (not shown is this work) showed no performance improvements regarding reduction of asymmetric loads, e.g. yaw drive, and further modeling details added to the control design model are needed before individual blade pitch can be enabled with success.

Model predictive control in a relinearization framework has been used to control the floating wind turbine. The relinearization scheme enables the linearized controller to always be linearized around the current operating point leading to better control performance.

An extended Kalman filter has been used to estimated the unmeasurable states e.g. wind speed and wave forces using only realistically available sensors.

Simulations in partial, partial/full and full load operation showed performance improvements compared to a benchmark PI controller and smooth transitions between partial full load operation has been achieved with the presented controllers.

The importance of including dynamic inflow in the control design model used by the state estimator and control algorithm has been investigated by comparing two controllers: The first assuming quasi-steady aerodynamics and the second including dynamic inflow. No significant differences between the two controllers were observed, except for the simulation with a mean wind speed of 12 m/s where the controller including dynamic inflow showed better performance than the controller assuming quasi-steady aerodynamics. Further investigations are needed to determine whether or not it is beneficial to include dynamic inflow in the control design model. Perhaps individual blade pitch operation will clarify the issue or simulations with a non-floating wind turbine where the controllers can be tuned more aggressive.

REFERENCES

1. Jonkman J. Dynamics modeling and loads analysis of an offshore floating wind turbine. *Technical Report NREL/TP-500-41958*, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401-3393 November 2007.
2. Nielsen FG, Hanson TD, Skaare B. Integrated dynamic analysis of floating wind turbines. *Proceedings of OMAE2006*, 2006.
3. Larsen TJ, Hanson TD. A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine. *J. Phys.: Conf. Ser.* 2007; **75 012073 (11pp)**:doi:10.1088/1742–6596/75/1/012 073.
4. Skaare B, Hanson TD, Nielsen FG. Importance of control strategies on fatigue life of floating wind turbines. *ASME Conference Proceedings* 2007; **2007**(42711):493–500, doi:10.1115/OMAE2007-29277.
5. Namik H, Stol K. Individual blade pitch control of floating offshore wind turbines. *Wind Energ.* 2010; **13**(1):74–85.
6. Henriksen LC, Hansen MH, Poulsen NK. Wind turbine control with contraint handling. *Submitted for Wind Energy* 2010; .
7. Kumar AA, Stol DK. Scheduled model predictive control of a wind turbine. *Proceedings of the European Wind Energy Conference 2009*, Warsaw, Poland, 2009.
8. Trainelli L, Sirchi W, Savini B, Croce A, Bottasso CL. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody System Dynamics* 2006; **16**(3):291–308.
9. Santos R. Damage mitigating control for wind turbines. PhD Thesis, University of Colorado at Boulder, United States – Colorado 2007.
10. Østergaard K, Stoustrup J, Brath P. Linear parameter varying control of wind turbines covering both partial load and full load conditions. *International Journal of Robust and Nonlinear Control* 2009; **19**(1):92–116.
11. Henriksen LC, Hansen MH, Poulsen NK. Wind speed and wave force estimation for a floating wind turbine. *Submitted for Wind Energy* 2010; .
12. Henriksen LC, Hansen MH, Poulsen NK. The effect of dynamic inflow in free mean wind speed estimation. *Submitted for Wind Energy* 2010; .
13. Mann J. Wind field simulation. *Probabilistic Engineering Mechanics* 1998; **13**(4):269–282.
14. Kristensen L, Frandsen S. Model for power spectra of the blade of a wind turbine measured from the moving frame of reference. *Journal of Wind Engineering and Industrial Aerodynamics* 1982; **10**(2):249 – 262, doi:DOI: 10.1016/0167-6105(82)90067-8.
15. Passon P, Kuhn M, Butterfield S, Jonkman J, Camp T, Larsen T. Oc3-benchmark exercise of aero-elastic offshore wind turbine codes. *J. Phys., Conf. Ser. (UK)* 2007; **75**(1):–.
16. Henriksen LC, Poulsen NK. Energy-based nonlinear control of hydraulically actuated pitch-servo systems. *Proceedings of the European Wind Energy Conference 2009*, Marseille, France, 2009.
17. Hansen MH, e BSK. Servo-elastic dynamics of a hydraulic actuator pitching a blade with large deflections. *Journal of Physics: Conference Series* 2007; **75**(1):012 077.
18. Larsen T, Hansen M, Iov F. Generator dynamics in aeroelastic analysis and simulations. *Risø-R 1395*, Risø National Laboratory 2003.
19. Muske KR, Badgwell TA. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control* 2002; **12**(5):617–632.
20. Pannocchia G, Rawlings JB. Disturbance models for offset-free model-predictive control. *AIChE Journal* 2003; **49**(2):426–437.
21. Scokaert P, Rawlings J. Constrained linear quadratic regulation. *Automatic Control, IEEE Transactions on* 1998; **43**(8):1163 –1169.
22. Nocedal J, Wright SJ. *Numerical Optimization*. 2nd edn., Springer, 2006.
23. Arnold I WF, Laub A. Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proc. IEEE (USA)* 1984; **72**(12):1746–1754.

24. Henriksen LC, Poulsen NK. An online re-linearization scheme suited for model predictive or linear quadratic control. *IMM-Technical Report 2010-13*, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark 2010.
25. Larsen TJ, Hansen AM. How 2 hawc2, the user's manual. *Technical Report Risø-R-1597(ver. 3-1)(EN)*, Risø National Laboratory 2007.
26. ASTM International. *ASTM E1049 - 85(2005) Standard Practices for Cycle Counting in Fatigue Analysis* 2005.

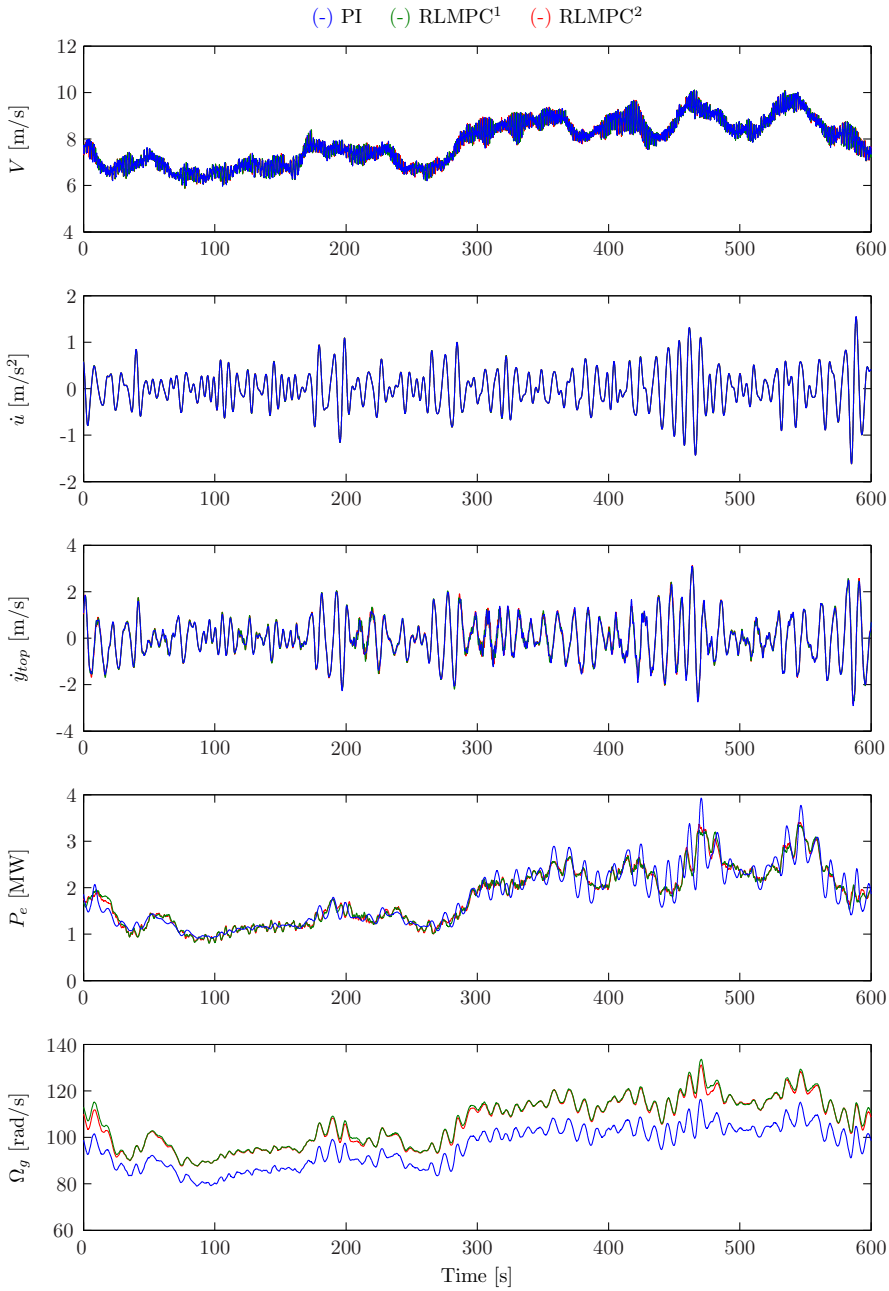


Figure 4. Wave forces (proportional to water acceleration $\dot{u}$) and turbulent wind with a mean speed of 8 m/s. Also depicted are tower top velocity, generator power and generator speed.

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

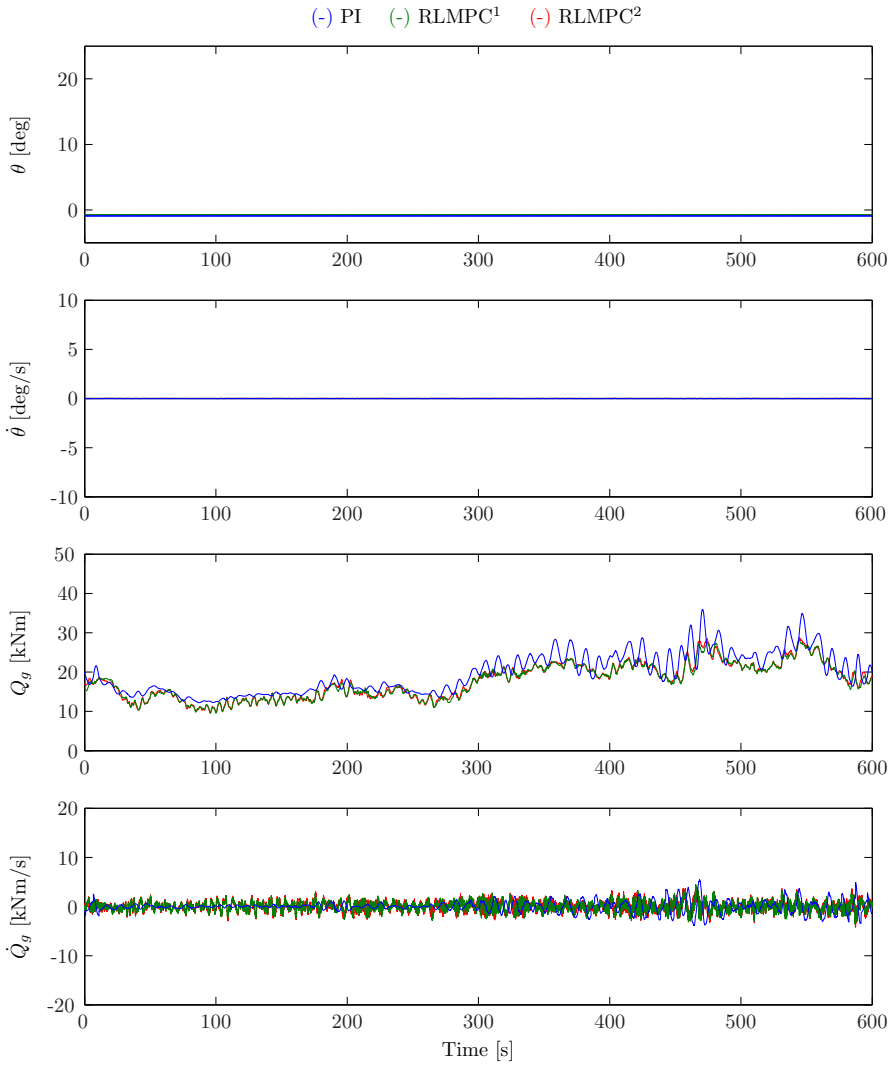


Figure 5. Control signals and rates at mean wind speed of 8 m/s.

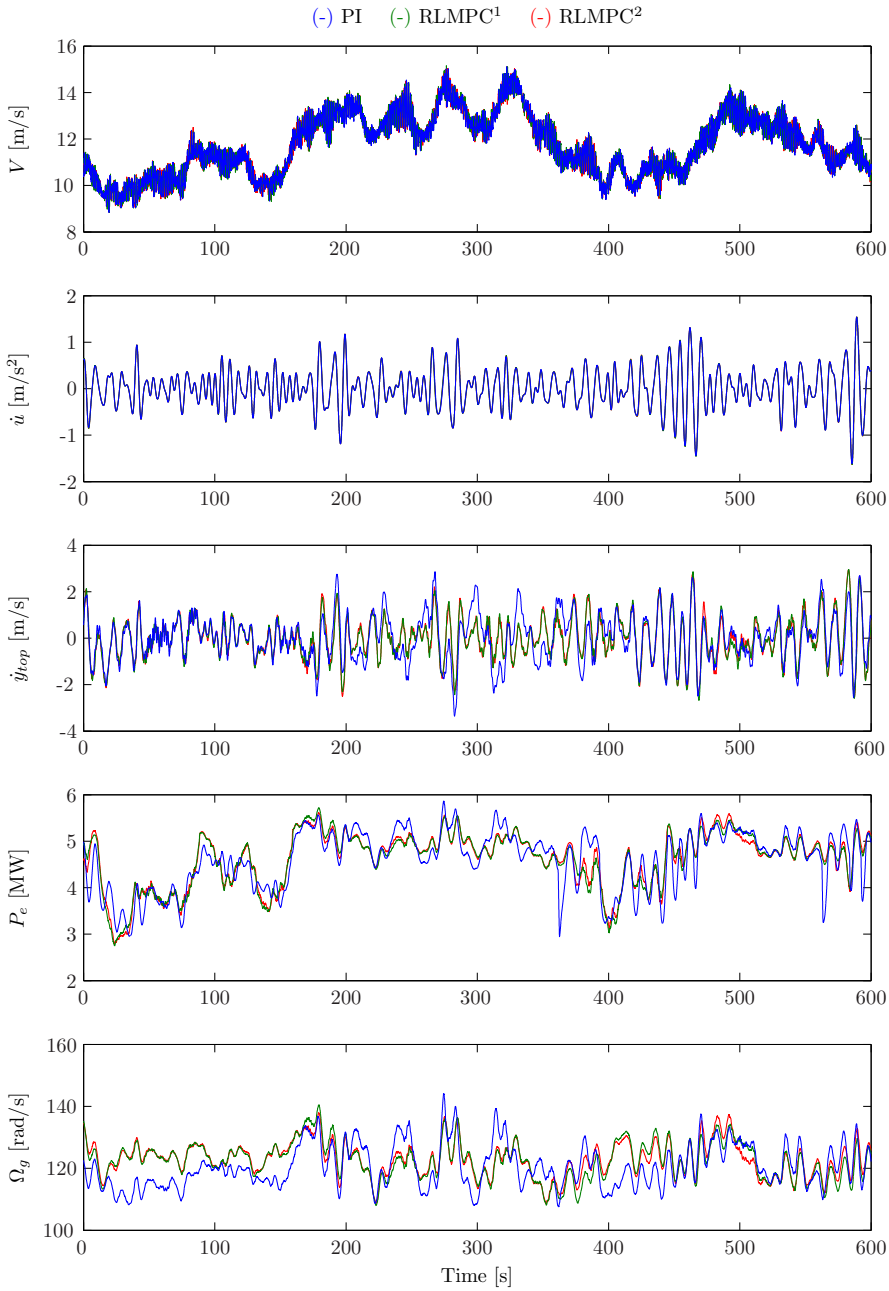


Figure 6. Wave forces (proportional to water acceleration $\dot{u}$) and turbulent wind with a mean speed of 12 m/s. Also depicted are tower top velocity, generator power and generator speed.

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

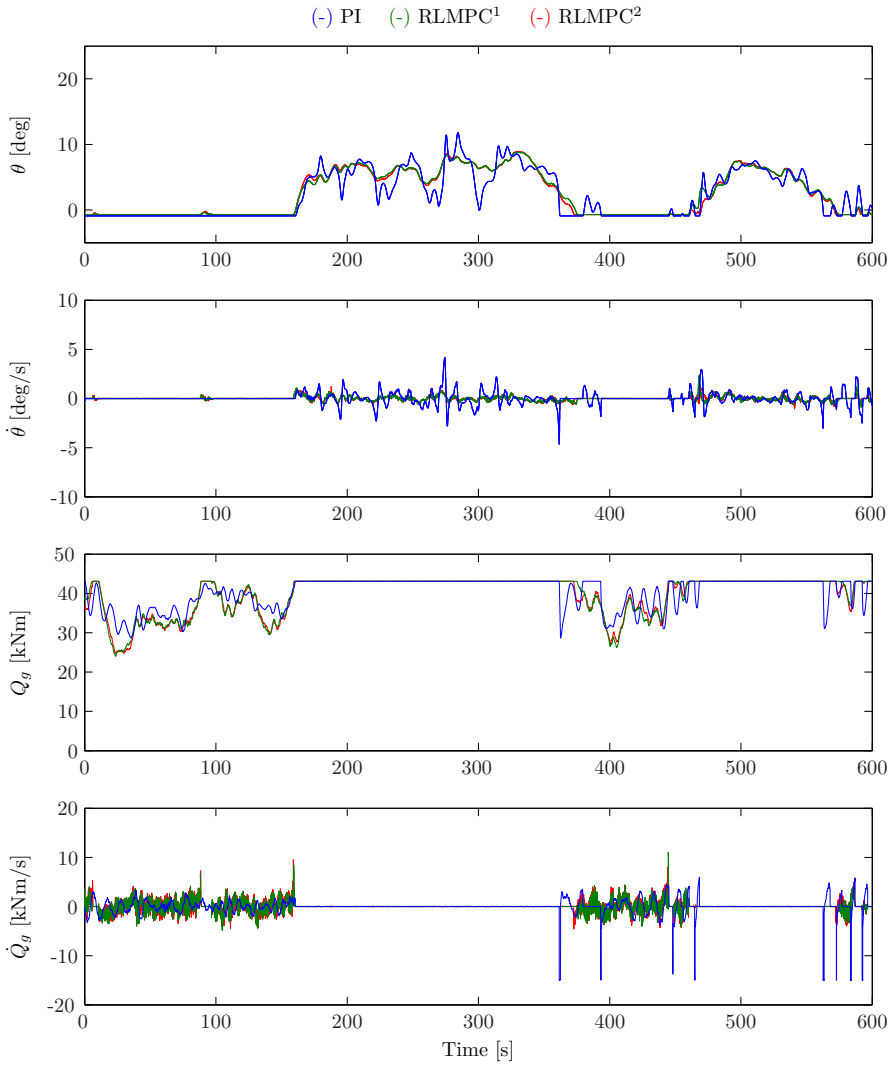


Figure 7. Control signals and rates at mean wind speed of 12 m/s.

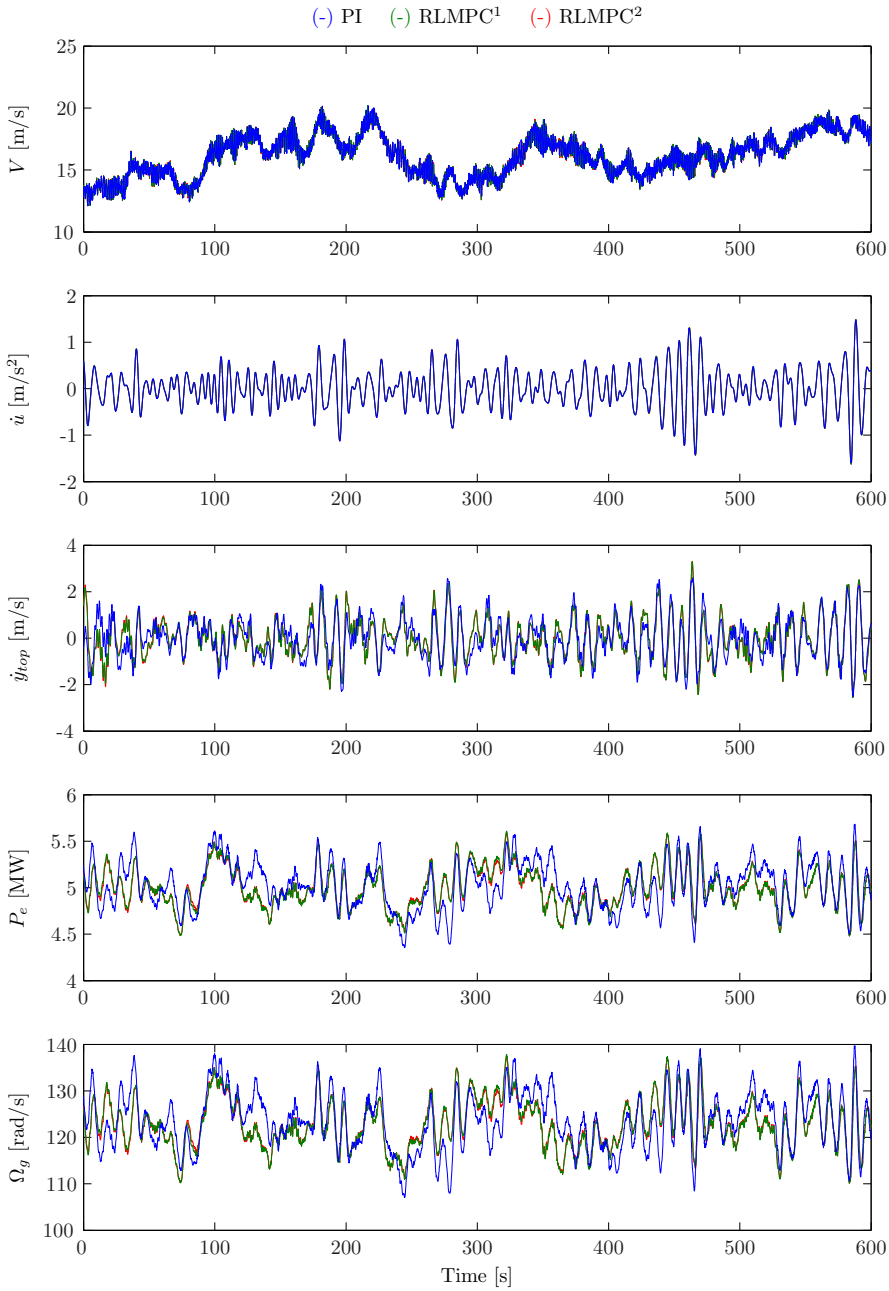


Figure 8. Wave forces (proportional to water acceleration $\dot{u}$) and turbulent wind with a mean speed of 16 m/s. Also depicted are tower top velocity, generator power and generator speed.

Wind Energ. 2010; **00**:1–15 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

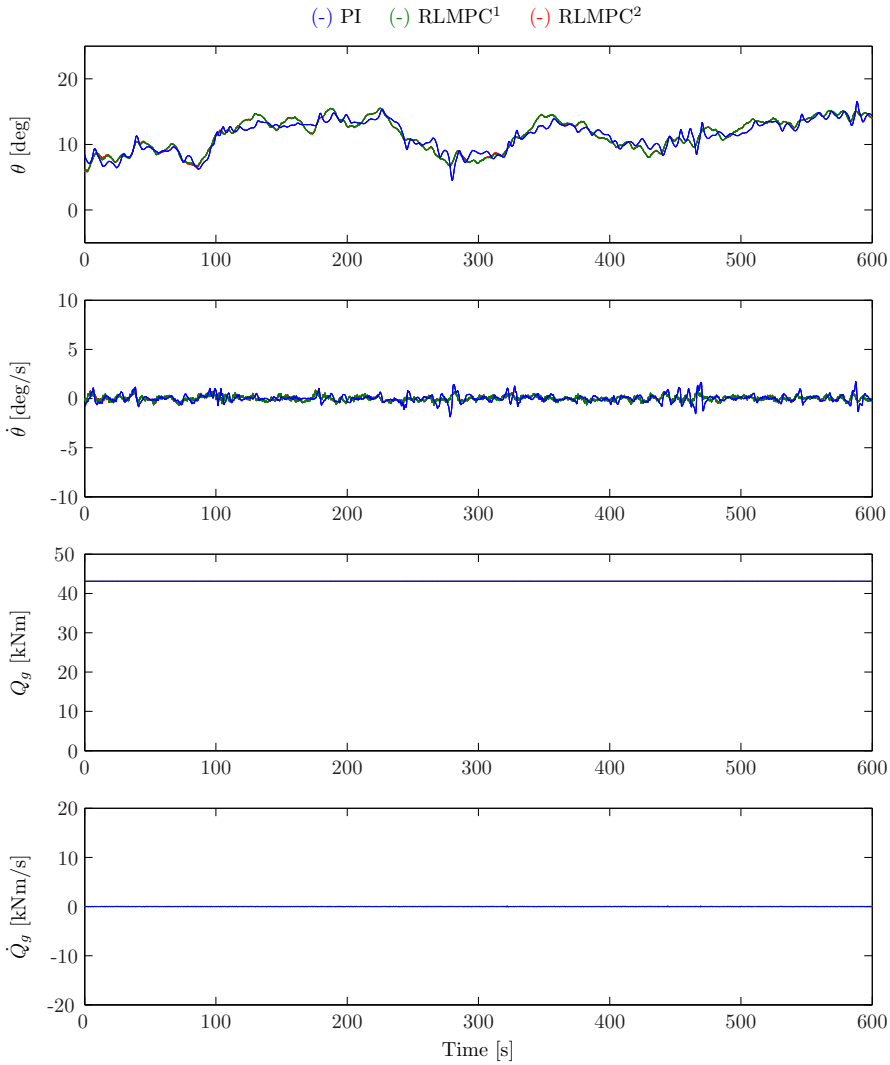


Figure 9. Control signals and rates at mean wind speed of 16 m/s.

Nonlinear Model Predictive Control of a Simplified Wind Turbine

L. C. Henriksen * **N. K. Poulsen** ** **M. H. Hansen** ***

* *Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, larh@risoe.dtu.dk*
** *Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, nkp@imm.dtu.dk*
*** *Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, mhha@risoe.dtu.dk*

Abstract: This paper discusses the implications of formulating a single control law governing the entire wind speed range of operation for a wind turbine. Furthermore, the knowledge of future wind speeds provided by e.g. LIDARs is included in the controller framework. This is possible as the presented controller is based on nonlinear model predictive control and includes the knowledge of the future wind speed in the prediction horizon of the controller. The potential benefits of exploiting the knowledge provided by LIDARs is demonstrated in simulations with a simplified 1 degree-of-freedom nonlinear wind turbine model.

Keywords: nonlinear model predictive control, constraint handling, horizontal axis wind turbine, lidar

1. INTRODUCTION

The variable-speed horizontal-axis pitch-controlled wind turbine [Burton et al., 2001] pose some interesting challenges with respect to the design of controllers. Partial load wind speeds, that is wind speeds where the power produced by the wind turbine is below the nominal generator power, the wind turbine controller seeks to maximize the wind power converted by the wind turbine. In this operating regime the wind turbine dynamics are highly nonlinear as the wind turbine operates on the top of the power coefficient (C_P) curve and depending on which side of the C_P curve the wind turbine currently operates the aerodynamic gradients switches sign when going from one side of the C_P curve to the other. For above rated wind speeds, the wind turbine has reached its nominal generator power and has pitched away from the top of the C_P curve. On the slope of the C_P curve, the wind turbine dynamics can be considered linear for reasonable small perturbations around the current operating point. Controllers usually switch between partial and full load configurations and the two configurations are usually quite different, where virtually no pitch action occurs for partial load operation and the electromagnetic generator torque attempts to keep the wind turbine at an optimal point of power capture, the roles reverse for full load operation where the generator torque action is reduced and the pitch control takes over.

There are good reasons to use pitch control below rated wind speeds as the spatial distribution of wind speeds seen in the area of the rotor disc is not uniform nor constant and load reductions on the key structures can be achieved by prober pitch action. Wake meandering, tower shadow and wind shear effects calls for individual pitch action, a subject not investigated further in this work. Another type of disturbance is rotor-wide wind speed changes, such as a extreme operating gust (EOG) [IEC/TC88, 2005], disturbances of that nature are the subject of interest for this work. The knowledge of future changes in wind speed can e.g. by achieved by the use of lidars [Angelou et al., 2010], by upstream meteorology masts or wind speed estimations of upstream wind turbines in a wind farm.

Nonlinear model predictive control (NMPC) [Qin and Badgwell, 2003] fit the needs for control of wind turbines with the knowledge of future wind speeds. NMPC is able to cope with the nonlinearities caused by operation below rated speeds and the knowledge of future wind speeds can be included in the prediction horizon of the NMPC. For wind speeds clearly in the partial or full load regions the NMPC controller can operate with a control law designed specifically for the particular region of operation. For wind speeds just around rated wind speed, the need for a single a control law is apparent. Initial thoughts of how to device such a control law are presented in this paper. NMPC has previously been applied for wind turbine control [Trainelli et al., 2006, Santos, 2007], but no special attention has been brought on below rated operation and on the switching between regions of operation.

The NMPC algorithm used in this work [Henriksen and Poulsen, 2010b,a] is partly based on the work of Tenny et al. [2004] and Rao et al. [1998] where the special structure of the MPC is exploited to reduce the computational burden of NMPC with long prediction horizons.

This paper is composed in the following way: The wind turbine model is presented first. The wind turbine model is followed by a section describing NMPC in general terms. The third section discusses relevant objectives for the wind turbine controller and discusses different options for implementing a single full wind speed range control law.

Finally results of simulations with the NMPC for above and below rated are presented and conclusions are made.

2. WIND TURBINE MODEL

The wind turbine used in this work is a 1 degree of freedom model with physical parameters similar to those of the NREL 5MW reference wind turbine defined by Jonkman et al. [2009]. A wind turbine of this type is controlled by the pitch angle θ of the blades and the electromagnetic torque of the generator Q_g, where generator torque in the one end of the drivetrain and aerodynamic rotor torque Q in the other determines whether the wind turbine rotational speed Ω is increased or decreased. The state space ordinary differential equation for the simplified wind turbine model, augmented with integrators between the control signals u_1 and u_2 and the pitch angle and generator torque, is

$$\begin{bmatrix} \dot{\Omega} \\ \dot{\theta} \\ \dot{Q}_g \end{bmatrix} = \begin{bmatrix} \frac{1}{J_t}\left(Q(V,\Omega,\theta) - Q_g\right) \\ u_1 \\ u_2 \end{bmatrix} \tag{1}$$

with the aerodynamic torque Q given by

$$Q(V,\Omega,\theta) = \frac{\frac{1}{2}\rho\pi R^2 V^3 C_P(V,\Omega,\theta)}{\Omega} \tag{2}$$

where ρ is mass density of air, R is the rotor radius and C_P is the power coefficient describing how much of the power available in the air is captured by the rotor. Generator power is given by

$$P_e = Q_g \Omega_g \eta, \quad \Omega_g = \Omega N_g \tag{3}$$

where η is the efficiency factor describing losses in gear, power electronics etc and N_g is the gear ratio. The aerodynamic power coefficient C_P can be mapped to be a function of (λ, θ) rather than (V, Ω, θ) where $\lambda = \Omega R/V$ as seen in Fig. 1(a).

The augmentation of the model, given by (1), with the two integrators is done to enable constraints on the control signal rates as well as on the control signals themselves.

3. NONLINEAR MODEL PREDICTIVE CONTROL

In this section the concept of Nonlinear Model Predictive Control will be presented. NMPC predicts the future behavior of the plant based on the model, current state estimates, and available knowledge of controlled and uncontrolled inputs. Controlled inputs are in this case the pitch angle and generator torque and the uncontrolled input is the wind speed. Typically no future knowledge of the disturbances is available, but with the naive assumption that e.g. a LIDAR is able to provide information about future wind speeds, this information can be used by the NMPC algorithm.

3.1 Constrained Dynamic Optimization

The control law is given by the minimization of the cost function Φ from current time t to the end of the prediction horizon at time $t + t_f$ with $N + 1$ discrete points within the finite prediction horizon

$$\min \Phi_N(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \Phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{s}_k, \boldsymbol{r}_k) \tag{4a}$$

subject to the following inequality constraints (4c) to (4e) and state progress constraint (4b)

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{d}_k) - \boldsymbol{x}_{k+1} = \boldsymbol{0} \tag{4b}$$

$$\boldsymbol{c}_h(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0} \tag{4c}$$

$$\boldsymbol{c}_s(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{s}_k \leq \boldsymbol{0} \tag{4d}$$

$$\boldsymbol{s}_k \geq \boldsymbol{0} \tag{4e}$$

where $\boldsymbol{x}$ are states, $\boldsymbol{u}$ are controlled inputs, $\boldsymbol{d}$ are uncontrolled inputs or disturbances, $\boldsymbol{r}$ are the references to be tracked and $\boldsymbol{s}$ are the slack variables associated with the soft constraint inequality. $\underline{\boldsymbol{f}}(\cdot)$ is the state progress equation, $\boldsymbol{c_h}(\cdot)$ is the hard constraints inequality and $\boldsymbol{c_s}(\cdot)$ is the soft constraints inequality. The total stage-wise cost function $\Phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{s}_k, \boldsymbol{r}_k)$ is the sum of the reference tracking cost (power control)

$$\Phi_r(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{r}_k) = \|\boldsymbol{y}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{r}_k\|^2_{\mathbf{W_r}} \tag{5}$$

the dynamic cost (load reduction)

$$\Phi_z(\boldsymbol{x}_k, \boldsymbol{u}_k) = \|\boldsymbol{z}(\boldsymbol{x}_k, \boldsymbol{u}_k)\|^2_{\mathbf{W_z}} \tag{6}$$

and of the cost of violating soft constraints

$$\Phi_s(\boldsymbol{s}_k) = \|\boldsymbol{s}_k\|^2_{\mathbf{W_s}} \tag{7}$$

A terminal cost $\Phi_N(\boldsymbol{x}_N)$ can be appended to achieve closed loop stability [Chen and Allgower, 1998]. But as discussed later on, this is not possible for the below rated operation and has thus been omitted from the controller. The open-loop system is not always stable and closed-loop predictions could be used to stabilize the system predictions within the prediction horizon [Tenny et al., 2004]. Open-loop instabilities have not given cause to any concern in the results obtained so far and has therefore not been implemented in the present work.

3.2 Time-discretization: Non-equidistant spacing

To exploit the potential benefit of having the knowledge of future wind speeds obtained by e.g. LIDARs, prediction horizons matching the length of future available knowledge are needed. It becomes beneficial w.r.t. computational costs to reduce the number of time-discrete points in the prediction horizon for long prediction horizons. The reduction of temporal points can be achieved by having a fine temporal resolution in the beginning of the prediction horizon and a coarser resolution towards the end of the prediction horizon. A non-equidistant temporal distribution means that the cost function can no longer be considered the sum of a number of equally important cost and the cost at a given temporal point should be weighted according to the current temporal spacing. Trapezoidal integration of the costs w.r.t. the temporal points of the prediction horizon ensures a proper weighting of the different cost along the prediction horizon. The temporal integration of the state progress equation should also be done with the non-equidistant temporal spacing in mind. Time discretization of the state progress equation can be done with e.g. forward Euler, Runge-Kutta schemes such as ESDIRK Kristensen et al. [2004], collocation points Biegler [2007] etc. The nonlinear model within the NMPC algorithm is repeatedly called throughout the temporal integration within a time step, resulting in a lot of computationally expensive function calls. To ease the computational burden, the model can be assumed to be linear within a time step and time integration can be performed using zero-order-hold or linear forward Euler. For larger time steps

as seen in the end of the prediction horizon zero-order-hold did not give satisfactory results and forward Euler has as result been used throughout this work. The assumption of linear dynamics within a time step might lead to poor convergence as the dynamics might change a lot from one iteration to another, especially for large time steps, it remains to be investigated whether the assumption of linear dynamics within a time step leads to overall faster computations or not.

3.3 Steady state performance

One method to determine if the control objectives achieve the desired steady state for a given constant wind speed is to perform an optimization with a cost function as one of the stage-wise costs from the dynamic optimization

$$\min_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{s}}} \Phi_{tot}(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{s}},\bar{\boldsymbol{r}}) \tag{8a}$$

s.t.

$$\boldsymbol{c}_h(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}) \leq \boldsymbol{0} \tag{8b}$$

$$\boldsymbol{c}_s(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}) - \boldsymbol{s} \leq \boldsymbol{0} \tag{8c}$$

$$\boldsymbol{f}(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}) = \boldsymbol{0} \tag{8d}$$

where the prediction horizon has been omitted from the optimization and the state progress equation has been replaced with the ordinary differential equation for the model.

3.4 Optimization algorithm

A trust-region-based sequential quadratic programming algorithm Henriksen and Poulsen [2010b] framework has been used as the optimization algorithm for both the steady state and for the dynamic optimization problems, where the underlying quadratic programming problem (QP) solvers, quasi-Newton Hessian approximations and trust-regions are specifically tailored for the specific problem. The steady state optimization uses a general purpose QP solver and produces dense quasi-Newton Hessian approximations and trust-regions. The dynamic optimization on the other hand use a QP solver exploiting the structure of the problem [Rao et al., 1998, Henriksen and Poulsen, 2010a] and quasi-Newton Hessian approximations and trust-regions also exploit the special structure of the problem as described by Tenny et al. [2004].

4. STEADY STATE CONTROL OBJECTIVES FOR THE WIND TURBINE

The first and foremost objective of the wind turbine control is to produce as much power as possible below rated wind speeds and to produce the nominal power above rated wind speeds. This objective can be formulated as

$$J = (P_e - P_{nom})^2 + w\,(\Omega_g - \Omega_{g,nom})^2 \tag{9}$$

where the additional term concerning generator speed Ω_g is only active for above rated wind speeds. This leads to $w = 0$ for partial load and $w = 1$ for full load. If the second term was omitted, then the generator speed could be a number of different values which together with a number of different pitch angles all lie on the same contour of the C_P curve. A soft constraint on generator speed, confining the generator speed to its allowed operating range, is active for $w = 0$. For $w = 1$ the second term is expected to keep generator speed within the allowed operating range.

A sweep of wind speeds ranging from V_{high} (= 25 m/s) to V_{low} (= 3 m/s) is investigated for steady state values. The sweep is started at the highest wind speed V_{high} and the initial guess is ensured to be on the pitch side of the C_P-curve. As steady state values for the initial wind speed is determined the second highest wind speed is investigated with the optimal values of the higher wind speed. Once partial load wind speeds are reached, the optimization is no longer able to determine values which ensure a power production at nominal power. The operating region weight w is changed to partial load and the sweep is continued for the below rated wind speeds. The mixed integer switching between one region of operation and another is not suitable for the dynamic optimization where regions of operation switch inside the prediction horizon. A smoother weight w depending on e.g. generator power should be designed and gradients for the gradients of w should be included in the gradients provided to optimization solver.

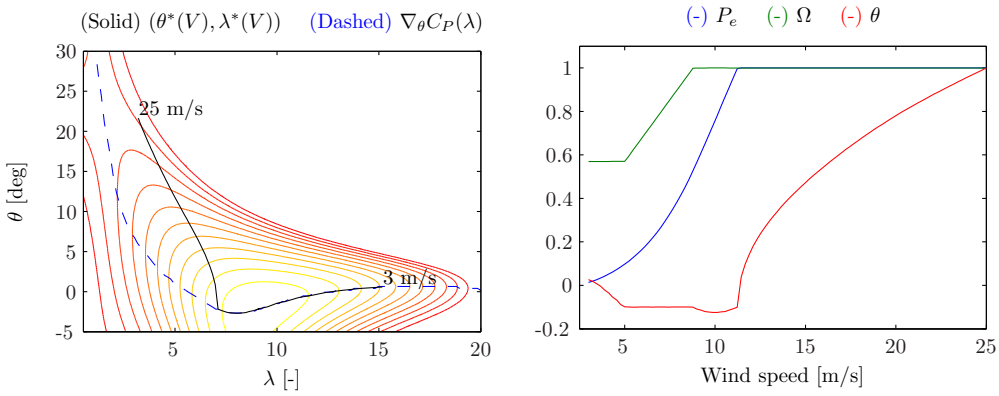
If the wind sweep goes from low to high wind speeds, a closer look at Fig. 1(a) shows that up until rated wind speed the wind turbine is operating on the ridge of the C_P curve. As rated wind speed is reached, power should be reduced and one of two options should taken: Either pitch to the stall side of the C_P curve or to the pitch side. The optimization algorithm has no preference to which option to choose and should be aided to ensure that solutions are found on the pitch side of the C_P curve rather than on the stall side. Nonlinear constraints could be imposed ensuring only feasible solutions on the pitch side another option is to use the fact that the optimization algorithm is gradient based and the aerodynamic gradient w.r.t. to pitch can be truncated such that increase in pitch angle always lead to decrease in C_P.

5. RESULTS

In this section results for the NMPC applied on a simplified wind turbine model is presented. First, partial load operation is examined and afterwards performance of full load operation is investigated. Two configurations of the NMPC is used: The first, NMPC$^{\text{lidar}}$, assumes that knowledge of future wind speeds is available within the entire prediction horizon. Whereas the second, NMPC$^{\text{normal}}$, has the same prediction horizon length as NMPC$^{\text{lidar}}$ but assumes the currently measured/estimated wind speed to remain constant throughout the prediction horizon.

5.1 Partial load operation

In this section partial load performance is investigated. No simulation results are presented as computation times where to long, only the finally iterated prediction horizons of the two controller configurations are shown in Fig. 2. The prediction horizon is extremely long, in this case 250 s. The necessity for a long prediction horizon can be seen by examination of the plots in Fig. 2 in the last approx. 50 s of the prediction horizon generator power is increased at the expense of generator speed. This behavior is as prescribed by the cost function (9) and to achieve good closed loop control the prediction horizon should be so



(a) C_P-curve (b) Primary variables

Fig. 1. Wind sweep for steady state operating points of the wind turbine: Fig. 1(a) depicts the C_P-curve and shows the steady state pitch angles and tip-speed-ratios for a sweep of wind speeds. Fig. 1(b) plots the normalized state values of generator power, generator speed and pitch angle.

long that the last 50 s does not influence the beginning of the prediction horizon. Another solution would be to append a suitable infinite horizon control law in the form of a terminal cost [Chen and Allgower, 1998]. The problem would however not be solved by appending a linear control law as the wind turbine is operating on the top of the C_P curve in partial load where the temporal dynamics of the wind turbine are approximated better by second order Taylor expanded models than by first order Taylor expanded models. Another option would be to change the control objectives in partial load operation from optimization of generator power to optimization of C_P, but different control objectives for partial and full load operation might lead to problems regarding convergence of the dynamic optimization problem when switching between the operating regions occur within the prediction horizon.

Another interesting observation, unfortunately not easily seen in Fig. 2, is that for the NMPC$^{\text{lidar}}$ the wind turbine is accelerated to higher rotational speed prior to the EOG. This is done to optimize the cost function and to produce more power than would have been produced if the rotational speed had not been accelerated. This observation indicates that also for more complex wind structures e.g. turbulence, the wind turbine would be able to produce more power at partial load operation if information of future wind speeds where available and if these predicted wind speeds where to be trusted.

5.2 Full load operation

In this section, results for the performance of the NMPC for full load operation is presented. An extreme operating gust occurs and as time progresses NMPC$^{\text{lidar}}$ the EOG moves into the prediction horizon of NMPC$^{\text{lidar}}$ and future control moves are planned accordingly, the outcome of the two controller configurations can be seen in Fig. 3. The pitch action of the two controllers is quite similar as shown in Fig. 3(b) but the resulting generator speed is better controlled when the information of future wind speed is included in the prediction horizon. The better controlled generator speed can be seen as an indication of the load reduction potential when applied on wind turbine models with more degrees of freedom.

6. CONCLUSION AND FUTURE WORK

A Nonlinear Model Predictive Control algorithm has been presented and its application on a wind turbine has been shown. Initial thought concerning the implementation of a full wind speed range control law has been expressed and results for the implemented controller has been shown both in partial and full load operation. It has been shown that inclusion of future wind speeds in the prediction horizon is beneficial and leads to increased power production for partial load operation and to decreased loads for full load operation.

It remains to be determined how to achieve good combined full and partial load operation within the prediction horizon. Different strategies for operating region switching exist: The operating region weight w could be made continuous and dependant of e.g. P_e in a smooth way to aid the optimization algorithm, that way the operating regions would not be clearly defined within the prediction horizon and as the optimization routine iterates the regions of operation change according to the current iterate, which might leave the optimization in limbo as the cost function is not well defined and convex. Another measure for determining which operating region should be active at a given point in time within the prediction horizon could be to make w dependant of the predicted wind speed for the given point in time in the prediction horizon, that way the iterations would not change w and the cost function would remain constant and better convergence properties might be achieved. Yet another option would be to make the region of operation dependant of a discrete set of rules for but as for the first option this would mean that the cost function could change from one iteration to the other and lead to poor convergence properties of the optimization algorithm. Future work should investigate these details.

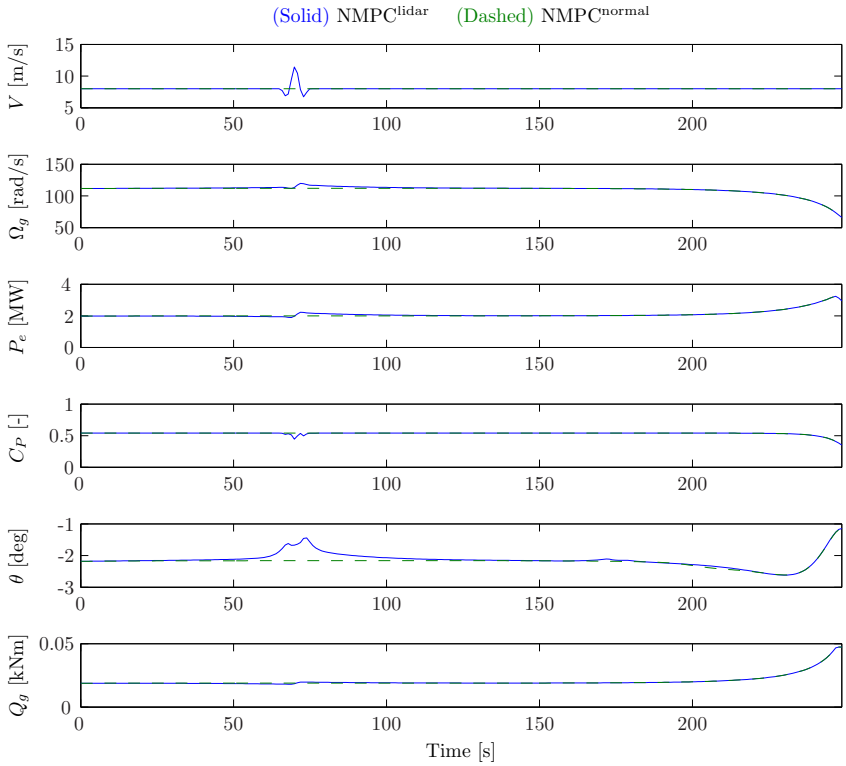


Fig. 2. Partial load operation: Prediction horizon of NMPC with no terminal cost. The plots show the predictions made by the controller at a given point in time.

BIBLIOGRAPHY

N. Angelou, T. Mikkelsen, K. H. Hansen, M. Sjöholm, and M. Harris. Lidar wind speed measurements from a rotating spinner: "spinnerex 2009". Risø-R 1741, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, 2010.

L. T. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process.: Process Intensif.*, 46(11):1043–1053, 2007.

T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi. *Wind Energy Handbook*. John Wiley And Sons Ltd, 2001.

H. Chen and F. Allgower. Quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

L. C. Henriksen and N. K. Poulsen. An online relinearization scheme suited for model predictive or linear quadratic control. IMM-Technical Report 2010-13, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, 2010a.

L. C. Henriksen and N. K. Poulsen. A trust-region-based sequential quadratic programming algorithm. IMM-Technical Report 2010-14, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, 2010b.

IEC/TC88. *IEC 61400-1 Ed.3: Wind turbines - Part 1: Design requirements*. International Electrotechnical Commission (IEC), 8 2005.

J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401-3393, February 2009.

M. R. Kristensen, J. B. Jørgensen, P. G. Thomsen, and S. B. Jørgensen. An esdirk method with sensitivity analysis capabilities. *Comput. Chem. Eng.*, 28(12): 2695–2707, 2004.

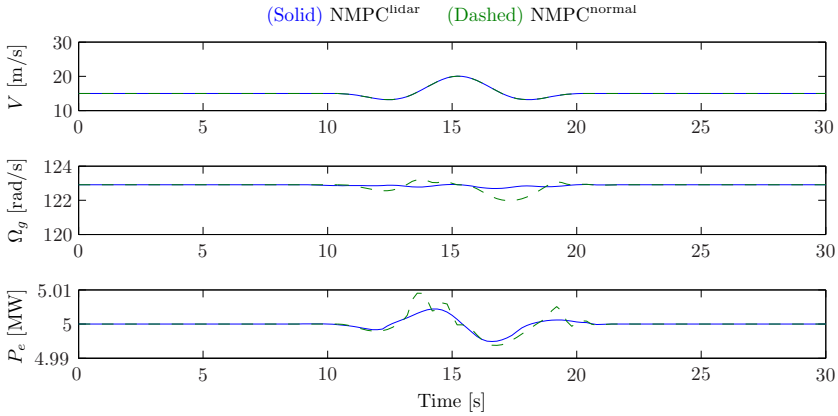
S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *J. Optim. Theory Appl. (USA)*, 99(3):723–757, 1998.

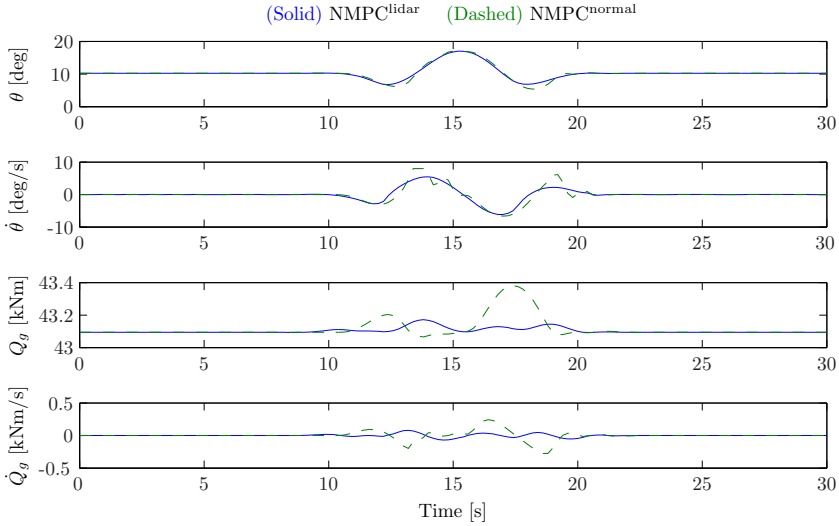
R. Santos. *Damage mitigating control for wind turbines*. PhD thesis, University of Colorado at Boulder, United States – Colorado, 2007.

M. J. Tenny, S. J. Wright, and J. B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, 2004.

L. Trainelli, W. Sirchi, B. Savini, A. Croce, and C. L. Bottasso. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody System Dynamics*, 16(3):291–308, 2006.



(a) Wind speed, generator speed and power.



(b) Controlled inputs to plant: Pitch angle and rate, and generator torque and rate.

Fig. 3. Above rated wind speed: Simulations where an extreme operating gust occurs. Two nonlinear model predictive controllers are compared: The first, NMPC$^{\text{lidar}}$, includes knowledge of future wind speeds in the prediction horizon. The second, NMPC$^{\text{normal}}$, assumes that the current wind speed remains constant throughout the prediction horizon.

Part III

Numerical Methods

IMM - TECHINICAL REPORT - 2010-13 - REV. 3

An online re-linearization scheme suited for Model Predictive and Linear Quadratic Control

L. C. Henriksen[1] and N. K. Poulsen[2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark, larh@risoe.dtu.dk
[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, nkp@imm.dtu.dk

ABSTRACT

This technical note documents the equations for primal-dual interior-point quadratic programming problem solver used for MPC. The algorithm exploits the special structure of the MPC problem and is able to reduce the computational burden such that the computational burden scales with prediction horizon length in a linear way rather than cubic, which would be the case if the structure was not exploited. It is also shown how models used for design of model-based controllers, e.g. linear quadratic and model predictive, can be linearized both at equilibrium and non-equilibrium points, making the presented extension of the controller formulation equivalent to that of the extended Kalman filter compared to an ordinary Kalman filter.

KEYWORDS
linear quadratic control, model predictive control, primal-dual interior-point quadratic programming problem

1 Introduction

State-space control methods based on linear models such as linear quadratic control (LQ), model predictive control (MPC) etc. are typically based on linearizations around equilibrium points, where the control objectives seek to minimize the distance from the current point of operation to the equilibrium point. Gain scheduling frameworks for different operating conditions can be used to handle nonlinearities. The classical gain scheduling framework, consisting of a Jacobian linearization family as described by Rugh and Shamma [1], is comprised of a family of linearizations all done at equilibrium points depending on the chosen scheduling variable. The mentioned control methods all have the same shortcoming, linearizations around equilibrium points are not guaranteed to resemble the actual dynamics at the current operating point. More advanced gain scheduling methods such as linear parameter varying (LPV) methods enable linearizations within a finite parameter space and are not limited to a set of equilibrium points. With an increasing number of scheduling variables and with an increasing discretization of the scheduling variables, the dimensions of the parameter space quickly grows and renders the practical implementation close to impossible, especially if constraints on states, outputs and inputs are to be handled by the controller. The easy fix to avoid the explosion of dimensions in the parameter space is to re-linearize the model online at each sample time and design a controller based on the linearized model, taking non-equilibrium residuals of the linearization into account in the controller design. No theoretical guarantees concerning stability etc. can however be given to this ad-hoc procedure compared to the theoretically better founded LPV framework.

If one or more outputs are to be steered towards their desired reference values several methods exits: The first is to augment the system with an integrator for each reference tracked output and seek to minimize the integrated the error between the output and the reference value. For varying reference values or constraints on e.g. control signals or control signal rates, integrator wind-up can occur, giving rise to unwanted overshoot performance of the closed-loop. A second option is to use target calculation to determine the new steady state target values for states and control signals and use an origin-shifting controller framework, e.g. as proposed by Pannocchia and Rawlings [2].

In this note, it is shown how to gather reference tracking and relinearization at non-stationary points, into a unified formulation, which serves as an extension to the standard Linear Quadratic (LQ) formulation. Model Predictive Control (MPC) with quadratic cost functions is similar to the standard LQ control and the unified control formulation can be applied to MPC as well. Rao et al. [3] has shown how to ease the computational burden of MPC by exploiting the structure given by an inequality constrained optimal control problem. The method suggested by Rao et al. [3] is extended to accommodate the unified formulation presented in this paper. Similar algorithms exploiting the structure of the interior-point optimization problem have subsequently been implemented by others, e.g. Edlund et al. [4] and Haverbeke et al. [5] for MPC and Moving Horizon Estimation purposes, respectively. The extended MPC formulation is also able to act as the quadratic programming problem (QP) solving algorithm used by a sequential quadratic programming solver, when applied to Nonlinear Model Predictive Control (NMPC) as done by Tenny et al. [6].

This documents outline is: In the first section unconstrained MPC and reference-tracking non-stationary LQ control is presented. In the second section it is shown how to exploit the structure of an inequality constrained MPC adapted to the unified control framework when solving the QP. To relate the second section to a quadratic programming problem solver, theory for a general purpose primal-dual interior-point quadratic programming is revisited in the third section. In the second last section it is shown how to obtain time discrete dynamic models at non-stationary points. The last section shows how to incorporate control moves into the framework.

2 Unconstrained linear model predictive control

The finite horizon optimal control problem is given as

$$\min \phi_N(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{1a}$$

where the stagewise cost function

$$\phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \frac{1}{2} \left\| \boldsymbol{r} - \boldsymbol{g}_{\boldsymbol{r}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \right\|^2_{\mathbf{W_r}} + \frac{1}{2} \left\| \boldsymbol{g}_{\boldsymbol{z}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \right\|^2_{\mathbf{W_z}} \tag{1b}$$

consist of two terms: The first term seeks to drive the plant reference outputs $\boldsymbol{g}_{\boldsymbol{r}}(\cdot)$ towards the desired reference $\boldsymbol{r}$. The second term seeks to minimize dynamic variations given by $\boldsymbol{g}_{\boldsymbol{z}}(\cdot)$ such as e.g. velocities, accelerations. The optimization problem is subject to an initial constraint

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}} \tag{1c}$$

and to the nonlinear state progress equation constraint

$$\boldsymbol{x}_{k+1} = \underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k), \text{ for } k = 0, \ldots, N-1 \tag{1d}$$

The nonlinear equations can be assumed linear throughout the entire prediction horizon to ease the problem solving. The state progress equation as well as the equations for the reference outputs $\boldsymbol{g}_{\boldsymbol{r}}(\cdot)$ and the dynamic outputs $\boldsymbol{g}_{\boldsymbol{z}}(\cdot)$ are linearized around $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \underline{\mathbf{A}}\boldsymbol{x}_k + \underline{\mathbf{B}}\boldsymbol{u}_k + \underline{\boldsymbol{\delta}} \tag{2a}$$

$$\boldsymbol{g}_{\boldsymbol{r}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{C_r}\boldsymbol{x}_k + \mathbf{D_r}\boldsymbol{u}_k + \boldsymbol{\gamma}_{\boldsymbol{r}} \tag{2b}$$

$$\boldsymbol{g}_{\boldsymbol{z}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{C_z}\boldsymbol{x}_k + \mathbf{D_z}\boldsymbol{u}_k + \boldsymbol{\gamma}_{\boldsymbol{z}} \tag{2c}$$

The linearized stagewise cost function can be put on a more general form

$$\phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \frac{1}{2}(\boldsymbol{x}_k^T \mathbf{Q} \boldsymbol{x}_k + \boldsymbol{u}_k^T \mathbf{R} \boldsymbol{u}_k + 2\boldsymbol{x}_k^T \mathbf{M} \boldsymbol{u}_k + 2\boldsymbol{q}^T \boldsymbol{x}_k + 2\boldsymbol{r}^T \boldsymbol{u}_k) \tag{3}$$

where

$$\mathbf{Q} = {\mathbf{C_r}}^T \mathbf{W_r} \mathbf{C_r} + {\mathbf{C_z}}^T \mathbf{W_z} \mathbf{C_z} \tag{4a}$$
$$\mathbf{R} = {\mathbf{D_r}}^T \mathbf{W_r} \mathbf{D_r} + {\mathbf{D_z}}^T \mathbf{W_z} \mathbf{D_z} \tag{4b}$$
$$\mathbf{M} = {\mathbf{C_r}}^T \mathbf{W_r} \mathbf{D_r} + {\mathbf{C_z}}^T \mathbf{W_z} \mathbf{D_z} \tag{4c}$$
$$\boldsymbol{q}^T = [\boldsymbol{\gamma_r} - \boldsymbol{r}]^T \mathbf{W_r} \mathbf{C_r} + {\boldsymbol{\gamma_z}}^T \mathbf{W_z} \mathbf{C_z} \tag{4d}$$
$$\boldsymbol{r}^T = [\boldsymbol{\gamma_r} - \boldsymbol{r}]^T \mathbf{W_r} \mathbf{D_r} + {\boldsymbol{\gamma_z}}^T \mathbf{W_z} \mathbf{D_z} \tag{4e}$$

and the final state cost

$$\phi_N(\boldsymbol{x}_N) = \frac{1}{2}(\boldsymbol{x}_N^T \boldsymbol{\Pi}_N \boldsymbol{x}_N + 2\boldsymbol{\pi}_N^T \boldsymbol{x}_N) \tag{5}$$

The Lagrangian to optimal control problem is

$$\mathcal{L} = \phi_N(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\nu}_0^T(\bar{\boldsymbol{x}} - \boldsymbol{x}_0) + \sum_{k=0}^{N-1} \boldsymbol{\nu}_{k+1}^T \left(\underline{\mathbf{A}}\boldsymbol{x}_k + \underline{\mathbf{B}}\boldsymbol{u}_k + \underline{\boldsymbol{\delta}} - \boldsymbol{x}_{k+1}\right) \tag{6}$$

The optimal solution is given by $\nabla\mathcal{L} = 0$, which can be found by a recursion of the stagewise Lagrangian gradients

$$\nabla_{\boldsymbol{x}_k}\mathcal{L} = \mathbf{Q}\boldsymbol{x}_k + \mathbf{M}\boldsymbol{u}_k + \boldsymbol{q} + \underline{\mathbf{A}}^T\boldsymbol{\nu}_{k+1} - \boldsymbol{\nu}_k \tag{7}$$
$$\nabla_{\boldsymbol{u}_k}\mathcal{L} = \mathbf{M}^T\boldsymbol{x}_k + \mathbf{R}\boldsymbol{u}_k + \boldsymbol{r} + \underline{\mathbf{B}}^T\boldsymbol{\nu}_{k+1} \tag{8}$$

starting a the end of the prediction horizon

$$\nabla_{\boldsymbol{x}_N}\mathcal{L} = \boldsymbol{\Pi}_N + \boldsymbol{\pi}_N - \boldsymbol{\nu}_N \tag{9}$$

inserting the final stage Lagrangian gradient (9) and the state progress equation (2a) into the $N-1$ stage Lagrangian gradient yields

$$\nabla_{\boldsymbol{x}_{N-1}}\mathcal{L} = [\mathbf{Q}_{N-1} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\underline{\mathbf{A}}]\boldsymbol{x}_{N-1} + \tilde{\mathbf{M}}_{N-1}\boldsymbol{u}_{N-1} + \tilde{\boldsymbol{q}}_{N-1} - \boldsymbol{\nu}_{N-1} \tag{10}$$
$$\nabla_{\boldsymbol{u}_{N-1}}\mathcal{L} = \tilde{\mathbf{M}}_{N-1}^T\boldsymbol{x}_{N-1} + \tilde{\mathbf{R}}_{N-1}\boldsymbol{u}_{N-1} + \tilde{\boldsymbol{r}}_{N-1} \tag{11}$$

where

$$\tilde{\mathbf{R}}_{N-1} = \mathbf{R} + \underline{\mathbf{B}}^T\boldsymbol{\Pi}_N\underline{\mathbf{B}} \text{ and } \tilde{\mathbf{M}}_{N-1} = \mathbf{M} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\underline{\mathbf{B}} \tag{12}$$

and

$$\tilde{\boldsymbol{q}}_{N-1} = \boldsymbol{q} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\underline{\boldsymbol{\delta}} + \underline{\mathbf{A}}^T\boldsymbol{\pi}_N \text{ and } \tilde{\boldsymbol{r}}_{N-1} = \boldsymbol{r} + \underline{\mathbf{B}}^T\boldsymbol{\Pi}_N\underline{\boldsymbol{\delta}} + \underline{\mathbf{B}}^T\boldsymbol{\pi}_N \tag{13}$$

are introduced to simplify notation. A control law can be derived from (11)

$$\boldsymbol{u}_{N-1} = -\mathbf{K}_{N-1}\boldsymbol{x}_{N-1} - \boldsymbol{\kappa}_{N-1} \tag{14}$$

where

$$\mathbf{K}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1}\tilde{\mathbf{M}}_{N-1}^T \text{ and } \boldsymbol{\kappa}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1}\tilde{\boldsymbol{r}}_{N-1} \tag{15}$$

which can be inserted into (10) giving

$$\nabla_{\boldsymbol{x}_{N-1}}\mathcal{L} = \boldsymbol{\Pi}_{N-1} + \boldsymbol{\pi}_{N-1} - \boldsymbol{\nu}_{N-1} \tag{16}$$

where

$$\boldsymbol{\Pi}_{N-1} = \mathbf{Q} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\underline{\mathbf{A}} - \tilde{\mathbf{M}}_{N-1}\mathbf{K}_{N-1} \text{ and } \boldsymbol{\pi}_{N-1} = \tilde{\boldsymbol{q}}_{N-1} - \tilde{\mathbf{M}}_{N-1}\boldsymbol{\kappa}_{N-1} \tag{17}$$

The recursion can then be continued for the stages $N-2$, $N-3$ etc. until the beginning of the prediction horizon. The variables can be reconstructed, if needed, with a forward recursion initiated with

$$\bar{\boldsymbol{x}} = \boldsymbol{x}_0 \tag{18}$$
$$\boldsymbol{\nu}_0 = \boldsymbol{\Pi}_0 \boldsymbol{x}_0 + \boldsymbol{\pi}_0 \tag{19}$$

and then going from $k = 0, \ldots, N-1$ where

$$\boldsymbol{u}_k = -\mathbf{K}_k \boldsymbol{x}_k - \boldsymbol{\kappa}_k \tag{20}$$
$$\boldsymbol{x}_{k+1} = \underline{\mathbf{A}} \boldsymbol{x}_k + \underline{\mathbf{B}} \boldsymbol{u}_k + \underline{\boldsymbol{\delta}} \tag{21}$$
$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\Pi}_{k+1} \boldsymbol{x}_{k+1} + \boldsymbol{\pi}_{k+1} \tag{22}$$

For a LTI system and $N \to \infty$ the solution becomes an algebraic set of equations. The quadratic cost for the terminal cost is found by the discrete-time algebraic Riccati equation (DARE)

$$\boldsymbol{\Pi} = \mathbf{Q} + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \underline{\mathbf{A}} - \tilde{\mathbf{M}} \mathbf{K} \tag{23}$$

where to simplify notation, the matrices

$$\tilde{\mathbf{M}} = \mathbf{M} + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \underline{\mathbf{B}} \text{ and } \tilde{\mathbf{R}} = \mathbf{R} + \underline{\mathbf{B}}^T \boldsymbol{\Pi} \underline{\mathbf{B}}$$

are introduced, leading to the feedback gain $\mathbf{K} = \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{M}}^T$. The linear terminal cost term $\boldsymbol{\pi}$ is determined from the same recursion as the DARE

$$\boldsymbol{\pi} = \boldsymbol{q} - \tilde{\mathbf{M}} \boldsymbol{\kappa} + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \underline{\boldsymbol{\delta}} + \underline{\mathbf{A}}^T \boldsymbol{\pi} \tag{24}$$

where the control action contribution $\boldsymbol{\kappa}$

$$\boldsymbol{\kappa} = \tilde{\mathbf{R}}^{-1} [\boldsymbol{r} + \underline{\mathbf{B}}^T \boldsymbol{\Pi} \underline{\boldsymbol{\delta}} + \underline{\mathbf{B}}^T \boldsymbol{\pi}] \tag{25}$$

is part of the optimal control law

$$\boldsymbol{u}_k = -\mathbf{K} \boldsymbol{x}_k - \boldsymbol{\kappa} \tag{26}$$

The DARE should be solved using a specialized DARE solver [7] to achieve fast and robust results and the linear cost term can be found from (24)

$$\boldsymbol{\pi} = [\mathbf{I} - \underline{\mathbf{A}}^T + \mathbf{K}^T \underline{\mathbf{B}}^T]^{-1} [\boldsymbol{q} - \mathbf{K}^T [\boldsymbol{r} + \underline{\mathbf{B}}^T \boldsymbol{\Pi} \underline{\boldsymbol{\delta}}] + \underline{\mathbf{A}}^T \boldsymbol{\Pi} \underline{\boldsymbol{\delta}}] \tag{27}$$

3 Inequality constrained MPC

In this section, the temporal subscript $_k$ has been omitted for the Jacobians $\underline{\mathbf{A}}, \underline{\mathbf{B}}$ etc. even though that they can vary within the prediction horizon, if used within a NMPC framework. This is done to ease notation and because they remain constant within the iterations of the QP solver.

The model predictive controller entails the computation of the control signal within a prediction horizon in the range $k = (0, \ldots, N-1)$. The MPC is formulated as a dual mode horizon where the first part, i.e $k = (0, \ldots, N-1)$, is considered constrained. In the second horizon, i.e. $k = (N + \ldots, \infty)$, it is assumed that the plant has reached a state where the unconstrained solution is feasible [8]. The dual mode optimization problem is

$$\min \sum_{k=0}^{N-1} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \sum_{k=1}^{N-1} \|\boldsymbol{\sigma}_k\|^2_{\mathbf{W}_\sigma} + \sum_{k=N}^{\infty} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{28a}$$

where the stagewise cost function is $\phi_k(\cdot)$ is given by (1b). The term $\|\boldsymbol{\sigma}_k\|^2_{\mathbf{W}_\sigma}$ is only included in the first part of the prediction horizon and seeks to minimize the violation of the soft constraints $\boldsymbol{\sigma}$. The optimization problem is subject to an initial constraint as in (1c) and to the state progress

equation (2a) in the interval ($k = 0, \ldots, \infty$). Whereas the soft and hard inequality constraints are only active in the first part of the prediction horizon

$$\boldsymbol{g_s}(\boldsymbol{x}_k) - \boldsymbol{\sigma}_k \approx \mathbf{C_s}\boldsymbol{x}_k + \boldsymbol{\gamma_s} - \boldsymbol{\sigma}_k \leq \boldsymbol{s}, \qquad k = (1, \ldots, N-1) \tag{28b}$$

$$\boldsymbol{\sigma}_k \geq 0, \qquad k = (1, \ldots, N-1) \tag{28c}$$

$$\boldsymbol{g_h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{C_h}\boldsymbol{x}_k + \mathbf{D_h}\boldsymbol{u}_k + \boldsymbol{\gamma_h} \leq \boldsymbol{h}, \qquad k = (0, \ldots, N-1) \tag{28d}$$

The constants contributions and the inequality limits can be combined in a redefined inequality limit to ease notation

$$\bar{\boldsymbol{s}} \equiv \boldsymbol{s} - \boldsymbol{\gamma_s} \text{ and } \bar{\boldsymbol{h}} \equiv \boldsymbol{h} - \boldsymbol{\gamma_h}$$

The second part of the optimization problem can be reduced to a terminal cost, consisting of a quadratic $\boldsymbol{\Pi}_N$ and a linear term $\boldsymbol{\pi}_N$

$$\phi_N(\boldsymbol{x}_N) = \sum_{k=N}^{\infty} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_N^T \boldsymbol{\Pi}_N \boldsymbol{x}_N + \boldsymbol{\pi}_N^T \boldsymbol{x}_N \tag{29}$$

given by the unconstrained DARE (23) and (27). The Lagrangian for the inequality constrained problem can written as

$$\begin{aligned}\mathcal{L} = \phi_N(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \sum_{k=1}^{N-1} \|\boldsymbol{\sigma}_k\|^2_{\mathbf{W}_\sigma} \\ + \boldsymbol{\nu}_0^T(\bar{\boldsymbol{x}} - \boldsymbol{x}_0) + \sum_{k=0}^{N-1} \left(\boldsymbol{\nu}_{k+1}^T(\underline{\mathbf{A}}\boldsymbol{x}_k + \underline{\mathbf{B}}\boldsymbol{u}_k + \underline{\boldsymbol{\delta}} - \boldsymbol{x}_{k+1})\right) \\ + \sum_{k=0}^{N-1} \left((\boldsymbol{\lambda}_k^{\boldsymbol{h}})^T(\mathbf{C_h}\boldsymbol{x}_k + \mathbf{D_h}\boldsymbol{u}_k - \bar{\boldsymbol{h}})\right) + \sum_{k=1}^{N-1} \left((\boldsymbol{\lambda}_k^{\boldsymbol{s}})^T(\mathbf{C_s}\boldsymbol{x}_k - \boldsymbol{\sigma}_k - \bar{\boldsymbol{s}})\right) + \sum_{k=1}^{N-1} \left((\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}})^T(-\boldsymbol{\sigma}_k)\right)\end{aligned} \tag{30}$$

where $\boldsymbol{\nu}_0$ is Lagrange multiplier for (1c); $\boldsymbol{\nu}_k$ for $k = 1, \ldots, N$ is the Lagrange multiplier for (2a); $\boldsymbol{\lambda}_k^{\boldsymbol{h}}$,$\boldsymbol{\lambda}_k^{\boldsymbol{s}}$ and $\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}}$ are the Lagrange multipliers for (28d),(28b) and (28c) respectively.

The Karush-Kuhn-Tucker (KKT) conditions for optimality are

$$\nabla\mathcal{L} = 0 \tag{31}$$

$$\bar{\boldsymbol{x}} - \boldsymbol{x}_0 = 0 \tag{32}$$

$$\underline{\mathbf{A}}\boldsymbol{x}_k + \underline{\mathbf{B}}\boldsymbol{u}_k + \underline{\boldsymbol{\delta}} - \boldsymbol{x}_{k+1} = 0 \tag{33}$$

$$\mathbf{C_h}\boldsymbol{x}_k + \mathbf{D_h}\boldsymbol{u}_k - \bar{\boldsymbol{h}} \leq 0 \tag{34}$$

$$\mathbf{C_s}\boldsymbol{x}_k - \boldsymbol{\sigma}_k - \bar{\boldsymbol{s}} \leq 0 \tag{35}$$

$$-\boldsymbol{\sigma}_k \leq 0 \tag{36}$$

$$\mathrm{diag}(\boldsymbol{\lambda}_k^{\boldsymbol{h}})\mathrm{diag}(\mathbf{C_h}\boldsymbol{x}_k + \mathbf{D_h}\boldsymbol{u}_k - \bar{\boldsymbol{h}})\boldsymbol{e} = 0 \tag{37}$$

$$\mathrm{diag}(\boldsymbol{\lambda}_k^{\boldsymbol{s}})\mathrm{diag}(\mathbf{C_s}\boldsymbol{x}_k - \boldsymbol{\sigma}_k - \bar{\boldsymbol{s}})\boldsymbol{e} = 0 \tag{38}$$

$$\mathrm{diag}(\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}})\mathrm{diag}(-\boldsymbol{\sigma}_k)\boldsymbol{e} = 0 \tag{39}$$

$$(\boldsymbol{\lambda}_k^{\boldsymbol{h}}, \boldsymbol{\lambda}_k^{\boldsymbol{s}}, \boldsymbol{\lambda}_k^{\boldsymbol{\sigma}}) \geq 0 \tag{40}$$

where (37), (38) and (39) are the complementary slackness conditions. By the introduction of the slack variables $\boldsymbol{t}_k^{\boldsymbol{h}}$, $\boldsymbol{t}_k^{\boldsymbol{s}}$ and $\boldsymbol{t}_k^{\boldsymbol{\sigma}}$ for $\boldsymbol{\lambda}_k^{\boldsymbol{h}}$, $\boldsymbol{\lambda}_k^{\boldsymbol{s}}$ and $\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}}$ respectively, the KKT condition can be rewritten

to

$$\mathcal{F} = \begin{bmatrix} \nabla\mathcal{L} \\ \bar{\boldsymbol{x}} - \boldsymbol{x}_0 \\ \underline{\mathbf{A}}\boldsymbol{x}_k + \underline{\mathbf{B}}\boldsymbol{u}_k + \underline{\boldsymbol{\delta}} - \boldsymbol{x}_{k+1} \\ \mathbf{C_h}\boldsymbol{x}_k + \mathbf{D_h}\boldsymbol{u}_k - \bar{\boldsymbol{h}} + \boldsymbol{t}_k^{\boldsymbol{h}} \\ \mathbf{C_s}\boldsymbol{x}_k - \boldsymbol{\sigma}_k - \bar{\boldsymbol{s}} + \boldsymbol{t}_k^{\boldsymbol{s}} \\ -\boldsymbol{\sigma}_k + \boldsymbol{t}_k^{\boldsymbol{\sigma}} \\ \mathbf{T}_k^{\boldsymbol{h}}\boldsymbol{\Lambda}_k^{\boldsymbol{h}}\boldsymbol{e} \\ \mathbf{T}_k^{\boldsymbol{s}}\boldsymbol{\Lambda}_k^{\boldsymbol{s}}\boldsymbol{e} \\ \mathbf{T}_k^{\boldsymbol{\sigma}}\boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}}\boldsymbol{e} \end{bmatrix} = 0 \tag{41}$$

$$(\boldsymbol{\lambda}_k^{\boldsymbol{h}}, \boldsymbol{t}_k^{\boldsymbol{h}}, \boldsymbol{\lambda}_k^{\boldsymbol{s}}, \boldsymbol{t}_k^{\boldsymbol{s}}, \boldsymbol{\lambda}_k^{\boldsymbol{\sigma}}, \boldsymbol{t}_k^{\boldsymbol{\sigma}}) \geq 0 \tag{42}$$

where there appropriate ranges of k are omitted for ease of notation. $\mathbf{T}_k^{\boldsymbol{h}} \equiv \text{diag}(\boldsymbol{t}_k^{\boldsymbol{\lambda}})$, $\boldsymbol{\Lambda}_k^{\boldsymbol{h}} \equiv \text{diag}(\boldsymbol{\lambda}_k^{\boldsymbol{h}})$, $\mathbf{T}_k^{\boldsymbol{s}} \equiv \text{diag}(\boldsymbol{t}_k^{\boldsymbol{s}})$, $\boldsymbol{\Lambda}_k^{\boldsymbol{s}} \equiv \text{diag}(\boldsymbol{\lambda}_k^{\boldsymbol{s}})$, $\mathbf{T}_k^{\boldsymbol{\sigma}} \equiv \text{diag}(\boldsymbol{t}_k^{\boldsymbol{\sigma}})$, $\boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}} \equiv \text{diag}(\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}})$ and $\boldsymbol{e} = [1\ 1 \ldots\ 1]^T$. The Jacobian of the Lagrangian $\nabla\mathcal{L}$ consist of

$$\nabla_{x_0}\mathcal{L} = \mathbf{Q}\boldsymbol{x}_0 + \mathbf{M}\boldsymbol{u}_0 + \boldsymbol{q}_0 - \boldsymbol{\nu}_0 + \underline{\mathbf{A}}^T\boldsymbol{\nu}_1 + \mathbf{C}_{\mathbf{h}}^T\boldsymbol{\lambda}_0^{\boldsymbol{h}} \tag{43}$$

$$\nabla_{x_k}\mathcal{L} = \mathbf{Q}\boldsymbol{x}_k + \mathbf{M}\boldsymbol{u}_k + \boldsymbol{q}_k - \boldsymbol{\nu}_k + \underline{\mathbf{A}}^T\boldsymbol{\nu}_{k+1} + \mathbf{C}_{\mathbf{h}}^T\boldsymbol{\lambda}_k^{\boldsymbol{h}} + \mathbf{C}_{\mathbf{s}}^T\boldsymbol{\lambda}_k^{\boldsymbol{s}} \tag{44}$$

$$\nabla_{x_N}\mathcal{L} = \boldsymbol{\Pi}_N\boldsymbol{x}_N + \boldsymbol{\pi}_N - \boldsymbol{\nu}_N \tag{45}$$

$$\nabla_{u_k}\mathcal{L} = \mathbf{R}\boldsymbol{u}_k + \mathbf{M}^T\boldsymbol{x}_k + \boldsymbol{r}_k + \underline{\mathbf{B}}^T\boldsymbol{\nu}_{k+1} + \mathbf{D}_{\mathbf{h}}^T\boldsymbol{\lambda}_k^{\boldsymbol{h}} \tag{46}$$

$$\nabla_{\epsilon_k}\mathcal{L} = \mathbf{Z}\boldsymbol{\sigma}_k + \boldsymbol{z} - \boldsymbol{\lambda}_k^{\boldsymbol{s}} - \boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} \tag{47}$$

The Newton like step

$$\nabla\mathcal{F}\Delta\boldsymbol{w} = -\mathcal{F} = \boldsymbol{r} \tag{48}$$

will be used to iterate towards a solution. The full KKT matrix $\nabla\mathcal{F}$ multiplied with the variable step $\Delta\boldsymbol{w} = [\Delta\boldsymbol{x}_0\ \Delta\boldsymbol{u}_0\ \ldots]$ and the residual vector $\boldsymbol{r} = [\boldsymbol{r}_0^{\boldsymbol{x}}\ \boldsymbol{r}_0^{\boldsymbol{u}}\ \ldots]$ are

$$\nabla\mathcal{F}\Delta w = \begin{bmatrix} \mathbf{Q}\Delta\boldsymbol{x}_0 + \mathbf{M}\Delta\boldsymbol{u}_0 - \Delta\boldsymbol{\nu}_0 + \underline{\mathbf{A}}^T\Delta\boldsymbol{\nu}_1 + \mathbf{C}_{\mathbf{h}}^T\Delta\boldsymbol{\lambda}_0^{\boldsymbol{h}} \\ \mathbf{Q}\Delta\boldsymbol{x}_k + \mathbf{M}\Delta\boldsymbol{u}_k - \Delta\boldsymbol{\nu}_k + \underline{\mathbf{A}}^T\Delta\boldsymbol{\nu}_{k+1} + \mathbf{C}_{\mathbf{h}}^T\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} + \mathbf{C}_{\mathbf{s}}^T\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} + \boldsymbol{q}_k \\ \boldsymbol{\Pi}_N\Delta\boldsymbol{x}_N - \Delta\boldsymbol{\nu}_N \\ \mathbf{R}\Delta\boldsymbol{u}_k + \mathbf{M}^T\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}^T\Delta\boldsymbol{\nu}_{k+1} + \mathbf{D}_{\mathbf{h}}^T\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} \\ \mathbf{Z}\Delta\boldsymbol{\sigma}_k - \Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} - \Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} \\ -\Delta\boldsymbol{x}_0 \\ \underline{\mathbf{A}}\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}\Delta\boldsymbol{u}_k - \Delta\boldsymbol{x}_{k+1} \\ \mathbf{D_h}\Delta\boldsymbol{u}_k + \mathbf{C_h}\Delta\boldsymbol{x}_k + \Delta\boldsymbol{t}_k^{\boldsymbol{h}} \\ \mathbf{C_s}\Delta\boldsymbol{x}_k - \Delta\boldsymbol{\sigma}_k + \Delta\boldsymbol{t}_k^{\boldsymbol{s}} \\ -\Delta\boldsymbol{\sigma}_k + \Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}} \\ \mathbf{T}_k^{\boldsymbol{h}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} + \boldsymbol{\Lambda}_k^{\boldsymbol{h}}\Delta\boldsymbol{t}_k^{\boldsymbol{h}} \\ \mathbf{T}_k^{\boldsymbol{s}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} + \boldsymbol{\Lambda}_k^{\boldsymbol{s}}\Delta\boldsymbol{t}_k^{\boldsymbol{s}} \\ \mathbf{T}_k^{\boldsymbol{\sigma}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} + \boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}}\Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}} \end{bmatrix} \tag{49}$$

$$\begin{bmatrix} \boldsymbol{r}_0^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{x}} \\ \boldsymbol{r}_N^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{u}} \\ \boldsymbol{r}_k^{\boldsymbol{\sigma}} \\ \boldsymbol{r}_0^{\boldsymbol{\nu}} \\ \boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} \\ \boldsymbol{r}_k^{\boldsymbol{\lambda}^{\boldsymbol{h}}} \\ \boldsymbol{r}_k^{\boldsymbol{\lambda}^{\boldsymbol{s}}} \\ \boldsymbol{r}_k^{\boldsymbol{\lambda}^{\boldsymbol{\sigma}}} \\ \boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{h}}} \\ \boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{s}}} \\ \boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{\sigma}}} \end{bmatrix} = -\mathcal{F} = \begin{bmatrix} -\mathbf{Q}\boldsymbol{x}_0 - \mathbf{M}\boldsymbol{u}_0 - \boldsymbol{q}_0 + \boldsymbol{\nu}_0 - \underline{\mathbf{A}}^T\boldsymbol{\nu}_1 - \mathbf{C}_\mathbf{h}^T\boldsymbol{\lambda}_0^{\boldsymbol{h}} \\ -\mathbf{Q}\boldsymbol{x}_k - \mathbf{M}\boldsymbol{u}_k - \boldsymbol{q}_k + \boldsymbol{\nu}_k - \underline{\mathbf{A}}^T\boldsymbol{\nu}_{k+1} - \mathbf{C}_\mathbf{h}^T\boldsymbol{\lambda}_k^{\boldsymbol{h}} - \mathbf{C}_\mathbf{s}^T\boldsymbol{\lambda}_k^{\boldsymbol{s}} \\ -\boldsymbol{\Pi}_N\boldsymbol{x}_N - \boldsymbol{\pi}_N + \boldsymbol{\nu}_N \\ -\mathbf{R}\boldsymbol{u}_k - \mathbf{M}^T\boldsymbol{x}_k - \boldsymbol{r}_k - \underline{\mathbf{B}}^T\boldsymbol{\nu}_{k+1} - \mathbf{D}_\mathbf{h}^T\boldsymbol{\lambda}_k^{\boldsymbol{h}} \\ -\mathbf{Z}\boldsymbol{\sigma}_k - \boldsymbol{z} + \boldsymbol{\lambda}_k^{\boldsymbol{s}} + \boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} \\ -\bar{\boldsymbol{x}} + \boldsymbol{x}_0 \\ -\underline{\mathbf{A}}\boldsymbol{x}_k - \underline{\mathbf{B}}\boldsymbol{u}_k - \underline{\boldsymbol{\delta}} + \boldsymbol{x}_{k+1} \\ -\mathbf{C}_\mathbf{h}\boldsymbol{x}_k - \mathbf{D}_\mathbf{h}\boldsymbol{u}_k + \bar{\boldsymbol{h}} - \boldsymbol{t}_k^{\boldsymbol{h}} \\ -\mathbf{C}_\mathbf{s}\boldsymbol{x}_k + \boldsymbol{\sigma}_k + \bar{\boldsymbol{s}} - \boldsymbol{t}_k^{\boldsymbol{s}} \\ \boldsymbol{\sigma}_k - \boldsymbol{t}_k^{\boldsymbol{\sigma}} \\ -\mathbf{T}_k^{\boldsymbol{h}}\boldsymbol{\Lambda}_k^{\boldsymbol{h}}\boldsymbol{e} - \Omega_h \\ -\mathbf{T}_k^{\boldsymbol{s}}\boldsymbol{\Lambda}_k^{\boldsymbol{s}}\boldsymbol{e} - \Omega_s \\ -\mathbf{T}_k^{\boldsymbol{\sigma}}\boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}}\boldsymbol{e} - \Omega_\sigma \end{bmatrix} \tag{50}$$

where $\Omega_h = \text{diag}(\Delta\boldsymbol{t}_k^{\boldsymbol{h}})\text{diag}(\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}}) - \sigma\mu\boldsymbol{e}$, $\Omega_s = \text{diag}(\Delta\boldsymbol{t}_k^{\boldsymbol{s}})\text{diag}(\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}}) - \sigma\mu\boldsymbol{e}$ and $\Omega_\sigma = \text{diag}(\Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}})\text{diag}(\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}}) - \sigma\mu\boldsymbol{e}$ are the centering terms for the corrector step in the predictor-corrector algorithm.

The slack variables $\Delta\boldsymbol{t}_k^{\boldsymbol{h}}$, $\Delta\boldsymbol{t}_k^{\boldsymbol{s}}$ and $\Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}}$ are eliminated using $(\Delta\boldsymbol{t}_k^{\boldsymbol{h}} = (\boldsymbol{\Lambda}_k^{\boldsymbol{h}})^{-1}(\boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{h}}} - \mathbf{T}_k^{\boldsymbol{h}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}})$, $(\Delta\boldsymbol{t}_k^{\boldsymbol{s}} = (\boldsymbol{\Lambda}_k^{\boldsymbol{s}})^{-1}(\boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{s}}} - \mathbf{T}_k^{\boldsymbol{s}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}})$ and $(\Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}} = (\boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}})^{-1}(\boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{\sigma}}} - \mathbf{T}_k^{\boldsymbol{\sigma}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}})$ respectively, giving

$$\nabla\mathcal{F}\Delta w = \begin{bmatrix} \mathbf{Q}\Delta\boldsymbol{x}_0 + \mathbf{M}\Delta\boldsymbol{u}_0 - \Delta\boldsymbol{\nu}_0 + \underline{\mathbf{A}}^T\Delta\boldsymbol{\nu}_1 + \mathbf{C}_\mathbf{h}^T\Delta\boldsymbol{\lambda}_0^{\boldsymbol{h}} \\ \mathbf{Q}\Delta\boldsymbol{x}_k + \mathbf{M}\Delta\boldsymbol{u}_k - \Delta\boldsymbol{\nu}_k + \underline{\mathbf{A}}^T\Delta\boldsymbol{\nu}_{k+1} + \mathbf{C}_\mathbf{h}^T\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} + \mathbf{C}_\mathbf{s}^T\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} \\ \boldsymbol{\Pi}_N\Delta\boldsymbol{x}_N - \Delta\boldsymbol{\nu}_N \\ \mathbf{R}\Delta\boldsymbol{u}_k + \mathbf{M}^T\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}^T\Delta\boldsymbol{\nu}_{k+1} + \mathbf{D}_\mathbf{h}^T\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} \\ \mathbf{Z}\Delta\boldsymbol{\sigma}_k - \Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} - \Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} \\ -\Delta\boldsymbol{x}_0 \\ \underline{\mathbf{A}}\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}\Delta\boldsymbol{u}_k - \Delta\boldsymbol{x}_{k+1} \\ \mathbf{D}_\mathbf{h}\Delta\boldsymbol{u}_k + \mathbf{C}_\mathbf{h}\Delta\boldsymbol{x}_k - \boldsymbol{\Sigma}_k^{\boldsymbol{h}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} \\ \mathbf{C}_\mathbf{s}\Delta\boldsymbol{x}_k - \Delta\boldsymbol{\sigma}_k - \boldsymbol{\Sigma}_k^{\boldsymbol{s}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} \\ -\Delta\boldsymbol{\sigma}_k - \boldsymbol{\Sigma}_k^{\boldsymbol{\sigma}}\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} \end{bmatrix} \tag{51}$$

where $\boldsymbol{\Sigma}_k^{\boldsymbol{h}} = (\boldsymbol{\Lambda}_k^{\boldsymbol{h}})^{-1}\mathbf{T}_k^{\boldsymbol{h}}$, $\boldsymbol{\Sigma}_k^{\boldsymbol{s}} = (\boldsymbol{\Lambda}_k^{\boldsymbol{s}})^{-1}\mathbf{T}_k^{\boldsymbol{s}}$ and $\boldsymbol{\Sigma}_k^{\boldsymbol{\sigma}} = (\boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}})^{-1}\mathbf{T}_k^{\boldsymbol{\sigma}}$

$$\begin{bmatrix} \boldsymbol{r}_0^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{x}} \\ \boldsymbol{r}_N^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{u}} \\ \boldsymbol{r}_k^{\boldsymbol{\sigma}} \\ \boldsymbol{r}_0^{\boldsymbol{\nu}} \\ \boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} \\ \bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{h}}} \\ \bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{s}}} \\ \bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{\sigma}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_0^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{x}} \\ \boldsymbol{r}_N^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{u}} \\ \boldsymbol{r}_k^{\boldsymbol{\sigma}} \\ \boldsymbol{r}_0^{\boldsymbol{\nu}} \\ \boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} \\ \boldsymbol{r}_k^{\boldsymbol{\lambda}^{\boldsymbol{h}}} - (\boldsymbol{\Lambda}_k^{\boldsymbol{h}})^{-1}\boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{h}}} \\ \boldsymbol{r}_k^{\boldsymbol{\lambda}^{\boldsymbol{s}}} - (\boldsymbol{\Lambda}_k^{\boldsymbol{s}})^{-1}\boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{s}}} \\ \boldsymbol{r}_k^{\boldsymbol{\lambda}^{\boldsymbol{\sigma}}} - (\boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}})^{-1}\boldsymbol{r}_k^{\boldsymbol{t}^{\boldsymbol{\sigma}}} \end{bmatrix} \tag{52}$$

Eliminating $\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}}$, $\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}}$ and $\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}}$ (using $\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} = (\boldsymbol{\Sigma}_k^{\boldsymbol{h}})^{-1}(\mathbf{D}_\mathbf{h}\Delta\boldsymbol{u}_k + \mathbf{C}_\mathbf{h}\Delta\boldsymbol{x}_k - \bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{h}}})$, $\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} = (\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}(\mathbf{C}_\mathbf{s}\Delta\boldsymbol{x}_k - \Delta\boldsymbol{\sigma}_k - \bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{s}}})$ and $\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} = (\boldsymbol{\Sigma}_k^{\boldsymbol{\sigma}})^{-1}(-\Delta\boldsymbol{\sigma}_k - \bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{\sigma}}})$), and afterwards eliminating $\Delta\boldsymbol{\sigma}_k$ (using $\Delta\boldsymbol{\sigma}_k = \bar{\mathbf{Z}}^{-1}(\mathbf{C}_\mathbf{s}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}\Delta\boldsymbol{x}_k + \bar{\boldsymbol{r}}_k^{\boldsymbol{\sigma}})$) where $\bar{\boldsymbol{r}}_k^{\boldsymbol{\sigma}} = \boldsymbol{r}_k^{\boldsymbol{\sigma}} - (\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}\bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{s}}} - (\boldsymbol{\Sigma}_k^{\boldsymbol{\sigma}})^{-1}\bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^{\boldsymbol{\sigma}}}$ yields

$$\nabla\mathcal{F}\Delta w = \begin{bmatrix} \bar{\mathbf{Q}}_0\Delta\boldsymbol{x}_0 + \bar{\mathbf{M}}_0\Delta\boldsymbol{u}_0 - \Delta\boldsymbol{\nu}_0 + \underline{\mathbf{A}}^T\Delta\boldsymbol{\nu}_1 \\ \bar{\mathbf{Q}}_k\Delta\boldsymbol{x}_k + \bar{\mathbf{M}}_k\Delta\boldsymbol{u}_k - \Delta\boldsymbol{\nu}_k + \underline{\mathbf{A}}^T\Delta\boldsymbol{\nu}_{k+1} \\ \boldsymbol{\Pi}_N\Delta\boldsymbol{x}_N - \Delta\boldsymbol{\nu}_N \\ \bar{\mathbf{R}}_k\Delta\boldsymbol{u}_k + \mathbf{M}_k^T\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}^T\Delta\boldsymbol{\nu}_{k+1} \\ -\Delta\boldsymbol{x}_0 \\ \underline{\mathbf{A}}\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}\Delta\boldsymbol{u}_k - \Delta\boldsymbol{x}_{k+1} \end{bmatrix} \tag{53}$$

and the residuals

$$\begin{bmatrix} \tilde{\boldsymbol{r}}_0^{\boldsymbol{x}} \\ \tilde{\boldsymbol{r}}_k^{\boldsymbol{x}} \\ \tilde{\boldsymbol{r}}_N^{\boldsymbol{x}} \\ \tilde{\boldsymbol{r}}_k^{\boldsymbol{u}} \\ \boldsymbol{r}_0^{\boldsymbol{\nu}} \\ \boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_0^{\boldsymbol{x}} + \mathbf{C}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_0^{\boldsymbol{h}})^{-1}\bar{\boldsymbol{r}}_0^{\boldsymbol{\lambda}^h} \\ \boldsymbol{r}_k^{\boldsymbol{x}} + \mathbf{C}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{h}})^{-1}\bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^h} + \mathbf{C}_{\mathbf{s}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}\bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^s} + \mathbf{C}_{\mathbf{s}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}\bar{\mathbf{Z}}_k^{-1}\bar{\boldsymbol{r}}_k^{\boldsymbol{\sigma}} \\ \boldsymbol{r}_N^{\boldsymbol{x}} \\ \boldsymbol{r}_k^{\boldsymbol{u}} + \mathbf{D}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{h}})^{-1}\bar{\boldsymbol{r}}_k^{\boldsymbol{\lambda}^h} \\ \boldsymbol{r}_0^{\boldsymbol{\nu}} \\ \boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} \end{bmatrix} \tag{54}$$

where

$$\bar{\mathbf{R}}_k = \mathbf{R} + \mathbf{D}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{h}})^{-1}\mathbf{D}_{\mathbf{h}} \tag{55}$$
$$\bar{\mathbf{M}}_k = \mathbf{M} + \mathbf{C}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{h}})^{-1}\mathbf{D}_{\mathbf{h}} \tag{56}$$
$$\bar{\mathbf{Z}}_k = \mathbf{Z} + (\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1} + (\boldsymbol{\Sigma}_k^{\boldsymbol{\sigma}})^{-1} \tag{57}$$
$$\bar{\mathbf{Q}}_0 = \mathbf{Q} + \mathbf{C}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_0^{\boldsymbol{h}})^{-1}\mathbf{C}_{\mathbf{h}} \tag{58}$$

where

$$\bar{\mathbf{Q}}_k = \mathbf{Q} - \mathbf{C}_{\mathbf{h}}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{h}})^{-1}\mathbf{C}_{\mathbf{h}} + \mathbf{C}_{\mathbf{s}}^T[(\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1} - (\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}\bar{\mathbf{Z}}_k^{-1}(\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}]\mathbf{C}_{\mathbf{s}} \tag{59}$$

using

$$\Delta\boldsymbol{u}_{N-1} = -\mathbf{K}_{N-1}\Delta\boldsymbol{x}_{N-1} + \boldsymbol{\kappa}_{N-1} \tag{60}$$

where

$$\tilde{\mathbf{R}}_{N-1} = [\bar{\mathbf{R}}_{N-1} + \underline{\mathbf{B}}^T\boldsymbol{\Pi}_N\underline{\mathbf{B}}] \tag{61}$$
$$\tilde{\mathbf{M}}_{N-1} = [\bar{\mathbf{M}}_{N-1} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\underline{\mathbf{B}}] \tag{62}$$
$$\mathbf{K}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1}\tilde{\mathbf{M}}_{N-1}^T \tag{63}$$
$$\boldsymbol{\Pi}_{N-1} = \bar{\mathbf{Q}}_{N-1} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\underline{\mathbf{A}} - \tilde{\mathbf{M}}_{N-1}\mathbf{K}_{N-1} \tag{64}$$
$$\boldsymbol{\kappa}_{N-1} = \tilde{\mathbf{R}}_{N-1}^{-1}[\tilde{\boldsymbol{r}}_{N-1}^{\boldsymbol{u}} + \underline{\mathbf{B}}^T\tilde{\boldsymbol{r}}_N^{\boldsymbol{x}} + \underline{\mathbf{B}}^T\boldsymbol{\Pi}_N\boldsymbol{r}_N^{\boldsymbol{\nu}}] \tag{65}$$
$$\boldsymbol{\pi}_{N-1} = \tilde{\boldsymbol{r}}_{N-1}^{\boldsymbol{x}} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_N\boldsymbol{r}_N^{\boldsymbol{\nu}} + \underline{\mathbf{A}}\tilde{\boldsymbol{r}}_N^{\boldsymbol{x}} - \tilde{\mathbf{M}}_{N-1}\boldsymbol{\kappa}_{N-1} \tag{66}$$

The recursion can be continued for $k = N-1, \ldots, 0$

$$\tilde{\mathbf{R}}_k = [\bar{\mathbf{R}}_k + \underline{\mathbf{B}}^T\boldsymbol{\Pi}_{k+1}\underline{\mathbf{B}}] \tag{67}$$
$$\tilde{\mathbf{M}}_k = [\bar{\mathbf{M}}_k + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_{k+1}\underline{\mathbf{B}}] \tag{68}$$
$$\mathbf{K}_k = \tilde{\mathbf{R}}_k^{-1}\tilde{\mathbf{M}}_k^T \tag{69}$$
$$\boldsymbol{\Pi}_k = \bar{\mathbf{Q}}_k + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_{k+1}\underline{\mathbf{A}} - \tilde{\mathbf{M}}_k\mathbf{K}_k \tag{70}$$
$$\boldsymbol{\kappa}_k = \tilde{\mathbf{R}}_k^{-1}[\tilde{\boldsymbol{r}}_k^{\boldsymbol{u}} + \underline{\mathbf{B}}^T\tilde{\boldsymbol{r}}_N^{\boldsymbol{x}} + \underline{\mathbf{B}}^T\boldsymbol{\Pi}_N\boldsymbol{r}_{k+1}^{\boldsymbol{\nu}}] \tag{71}$$
$$\boldsymbol{\pi}_k = \tilde{\boldsymbol{r}}_k^{\boldsymbol{x}} + \underline{\mathbf{A}}^T\boldsymbol{\Pi}_{k+1}\boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} + \underline{\mathbf{A}}\tilde{\boldsymbol{r}}_{k+1}^{\boldsymbol{x}} - \tilde{\mathbf{M}}_k\boldsymbol{\kappa}_k \tag{72}$$

A forward recursion can be used to construct variables

$$\Delta\boldsymbol{x}_0 = -\boldsymbol{r}_0^{\boldsymbol{\nu}} \tag{73}$$
$$\Delta\boldsymbol{\nu}_0 = \boldsymbol{\Pi}_0\Delta\boldsymbol{x}_0 - \boldsymbol{\pi}_0 \tag{74}$$

For $k = 0, \ldots, N-1$

$$\Delta\boldsymbol{u}_k = -\mathbf{K}_k\Delta\boldsymbol{x}_k + \boldsymbol{\kappa}_k \tag{75}$$
$$\Delta\boldsymbol{x}_{k+1} = \underline{\mathbf{A}}\Delta\boldsymbol{x}_k + \underline{\mathbf{B}}\Delta\boldsymbol{u}_k - \boldsymbol{r}_{k+1}^{\boldsymbol{\nu}} \tag{76}$$
$$\Delta\boldsymbol{t}_k^{\boldsymbol{h}} = -\mathbf{D}_{\mathbf{h}}\Delta\boldsymbol{u}_k - \mathbf{C}_{\mathbf{h}}\Delta\boldsymbol{x}_k + \boldsymbol{r}_k^{\boldsymbol{\lambda}^h} \tag{77}$$
$$\Delta\boldsymbol{\lambda}_k^{\boldsymbol{h}} = (\mathbf{T}_k^{\boldsymbol{h}})^{-1}(\boldsymbol{r}_k^{\boldsymbol{t}^h} - \boldsymbol{\Lambda}_k^{\boldsymbol{h}}\Delta\boldsymbol{t}_k^{\boldsymbol{h}}) \tag{78}$$

For $k = 1, \ldots, N-1$

$$\Delta\boldsymbol{\sigma}_k = \bar{\mathbf{Z}}^{-1}(\mathbf{C}_\mathbf{s}^T(\boldsymbol{\Sigma}_k^{\boldsymbol{s}})^{-1}\Delta\boldsymbol{x}_k + \bar{\boldsymbol{r}}_k^{\boldsymbol{\sigma}}) \tag{79}$$

$$\Delta\boldsymbol{\nu}_k = \boldsymbol{\Pi}_k\Delta\boldsymbol{x}_k - \boldsymbol{\pi}_k \tag{80}$$

$$\Delta\boldsymbol{t}_k^{\boldsymbol{s}} = -\mathbf{C}_\mathbf{s}\Delta\boldsymbol{x}_k + \Delta\boldsymbol{\sigma}_k + \boldsymbol{r}_k^{\boldsymbol{\lambda}^s} \tag{81}$$

$$\Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}} = \Delta\boldsymbol{\sigma}_k + \boldsymbol{r}_k^{\boldsymbol{\lambda}^\sigma} \tag{82}$$

$$\Delta\boldsymbol{\lambda}_k^{\boldsymbol{s}} = (\mathbf{T}_k^{\boldsymbol{s}})^{-1}(\boldsymbol{r}_k^{\boldsymbol{t}^s} - \boldsymbol{\Lambda}_k^{\boldsymbol{s}}\Delta\boldsymbol{t}_k^{\boldsymbol{s}}) \tag{83}$$

$$\Delta\boldsymbol{\lambda}_k^{\boldsymbol{\sigma}} = (\mathbf{T}_k^{\boldsymbol{\sigma}})^{-1}(\boldsymbol{r}_k^{\boldsymbol{t}^\sigma} - \boldsymbol{\Lambda}_k^{\boldsymbol{\sigma}}\Delta\boldsymbol{t}_k^{\boldsymbol{\sigma}}) \tag{84}$$

4 Linear constrained quadratic optimization

This section describes the underlying quadratic programming problem (QP) solver based on a interior-point primal-dual formulation. The algorithm (Alg. 1) is taken from Rao et al. [3], similar algorithms can be found in Wright [9] and Nocedal and Wright [10]. The mentioned algorithms are all based on the predictor-corrector method developed by Mehrotra [11]. The standard QP is given as

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \tag{85a}$$

subject to

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b} \tag{85b}$$

$$\mathbf{C}\boldsymbol{x} \leq \boldsymbol{d} \tag{85c}$$

where the Hessian $\mathbf{Q}$ is a symmetric positive semidefinite matrix. The Lagrangian to (85) is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{\lambda}^T(\mathbf{C}\boldsymbol{x} - \boldsymbol{d}) + \boldsymbol{\nu}^T(\mathbf{A}\boldsymbol{x} - \boldsymbol{b}) \tag{86}$$

Leading to the Karush-Kuhn-Tucker (KKT) conditions

$$\mathbf{Q}\boldsymbol{x} + \boldsymbol{c} + \mathbf{A}^T\boldsymbol{\nu} + \mathbf{C}^T\boldsymbol{\lambda} = 0 \tag{87a}$$

$$\mathbf{A}\boldsymbol{x} - \boldsymbol{b} = 0 \tag{87b}$$

$$\mathbf{C}\boldsymbol{x} - \boldsymbol{d} \leq 0 \tag{87c}$$

$$\boldsymbol{\lambda} \geq 0 \tag{87d}$$

$$\boldsymbol{\lambda}^T(\mathbf{C}\boldsymbol{x} - \boldsymbol{d})\boldsymbol{e} = 0 \tag{87e}$$

where (87a) is the stationarity condition, (87b) and (87c) are the primal feasibility conditions, (87d) is the dual feasibility condition and (87e) is the complimentary slackness condition for the primal-dual problem. Introducing the slack variable $\boldsymbol{t}$ to (85c) simplifies the KKT conditions and gives the following system to solved

$$\mathcal{F}(\boldsymbol{w}) = \begin{bmatrix} \mathbf{Q}\boldsymbol{x} + \boldsymbol{c} + \mathbf{A}^T\boldsymbol{\nu} + \mathbf{C}^T\boldsymbol{\lambda} \\ \mathbf{A}\boldsymbol{x} - \boldsymbol{b} \\ \mathbf{C}\boldsymbol{x} - \boldsymbol{d} - \boldsymbol{t} \\ \boldsymbol{\Lambda}^T\mathbf{T}\boldsymbol{e} \end{bmatrix} = 0, \text{ s.t. } (\boldsymbol{\lambda}, \boldsymbol{t}) \geq 0 \tag{88}$$

where $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$ and $\mathbf{T} = \text{diag}(\boldsymbol{t})$, $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{t})$. Newton like steps $\nabla\mathcal{F}\boldsymbol{\Delta w} = -\mathcal{F} = \boldsymbol{r}$ can be taken to iterate towards the solution. Multiple methods exist but one the most used is the predictor-corrector method by Mehrotra [11]. The method starts with a predictor step

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T & \mathbf{C}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{T} & \boldsymbol{\Lambda} \end{bmatrix} \begin{pmatrix} \boldsymbol{\Delta x}^{aff} \\ \boldsymbol{\Delta\nu}^{aff} \\ \boldsymbol{\Delta\lambda}^{aff} \\ \boldsymbol{\Delta t}^{aff} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{r}^{\boldsymbol{x}} \\ \boldsymbol{r}^{\boldsymbol{\nu}} \\ \boldsymbol{r}^{\boldsymbol{\lambda}} \\ \boldsymbol{r}^{\boldsymbol{t}} \end{pmatrix} \tag{89}$$

where

$$\boldsymbol{r}^{\boldsymbol{x}} = \mathbf{Q}\boldsymbol{x} + \boldsymbol{c} - \mathbf{A}^T\boldsymbol{\nu} - \mathbf{C}^T\boldsymbol{\lambda} \tag{90}$$

$$\boldsymbol{r}^{\boldsymbol{\nu}} = \mathbf{A}\boldsymbol{x} - \boldsymbol{b} \tag{91}$$

$$\boldsymbol{r}^{\boldsymbol{\lambda}} = \mathbf{C}\boldsymbol{x} - \boldsymbol{t} - \boldsymbol{d} \tag{92}$$

$$\boldsymbol{r}^{\boldsymbol{t}} = \boldsymbol{\Lambda}\mathbf{T}\boldsymbol{e} \tag{93}$$

The largest possible step size without violating the inequality constraints is calculated

$$\alpha^{aff} = \arg\max\{\alpha \in [0,1] | (\boldsymbol{\lambda}, \boldsymbol{s}) + \alpha(\boldsymbol{\Delta\lambda}^{aff}, \boldsymbol{\Delta s}^{aff}) \geq 0\} \tag{94}$$

The current complementary measure

$$\mu = \boldsymbol{\lambda}^T\boldsymbol{s}/\dim(\boldsymbol{\lambda}) \tag{95}$$

is a measure of the current feasibility and the affine complementary measure

$$\mu^{aff} = (\boldsymbol{\lambda} + \alpha^{aff}\boldsymbol{\Delta\lambda})^T(\boldsymbol{s} + \alpha^{aff}\boldsymbol{\Delta s})/\dim(\boldsymbol{\lambda}) \tag{96}$$

is a measure of the feasibility if the full predictor step was taken. The predictor step is followed by a corrector step which takes the centering parameter into account

$$\sigma = \left(\frac{\mu^{aff}}{\mu}\right)^3 \tag{97}$$

to keep the iterate at the central path between the primal and the dual problem and thus away from infeasibility. This is done by modifying the residual $\boldsymbol{r}^{\boldsymbol{t}}$ with the corrector term $\Omega = \mathrm{diag}(\boldsymbol{\Delta t}^{\boldsymbol{aff}})\mathrm{diag}(\boldsymbol{\Delta\lambda}^{\boldsymbol{aff}})\boldsymbol{e} - \sigma\mu\boldsymbol{e}$

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T & \mathbf{C}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{T} & \boldsymbol{\Lambda} \end{bmatrix} \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta\nu} \\ \boldsymbol{\Delta\lambda} \\ \boldsymbol{\Delta t} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{r}^{\boldsymbol{x}} \\ \boldsymbol{r}^{\boldsymbol{\nu}} \\ \boldsymbol{r}^{\boldsymbol{\lambda}} \\ \boldsymbol{r}^{\boldsymbol{t}} + \Omega \end{pmatrix} \tag{98}$$

The corrected step size can then be calculated

$$\alpha = \arg\max\{\alpha \in [0,1] | (\boldsymbol{\lambda}, \boldsymbol{s}) + \alpha(\boldsymbol{\Delta\lambda}, \boldsymbol{\Delta s}) \geq 0\} \tag{99}$$

and the variables can be updated

$$\boldsymbol{w}^+ = \boldsymbol{w} + \alpha\beta\boldsymbol{\Delta w}, \quad 0 < \beta < 1 \tag{100}$$

where β is a damping factor typically close to 1, this damping is imposed on the step to improve the numerical stability of the algorithm. Convergence is assumed if

$$\mu \leq \mathrm{tol}_\mu \text{ and } \|\boldsymbol{r}\|_\infty \leq \mathrm{tol}_r \|(\mathbf{Q}, \mathbf{A}, \mathbf{C}, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{d})\|_\infty \tag{101}$$

Care should also be taken when determining an initial guess, see e.g. Mehrotra [11] to determine good starting points for $\boldsymbol{\lambda}$ and $\boldsymbol{t}$ such that they are sufficiently far away from infeasibility but at the same time not to far away from each other as convergence speed would be impaired otherwise.

5 Linearization of models

A dynamic system can be described by the differential state equation and an output equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{102a}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{102b}$$

Algorithm 1: Interior-point primal-dual quadratic programming solver

```
Initial guess;
if Constraints are not violated then
    Return with result;
for ITER from 1 to IMAX do
    Solve ∇F Δw^aff = −r;
    Calc. α^aff = max{α ∈ (0,1] | (t, λ) + α(Δt^aff, Δλ^aff) ≥ 0} ;
    Calc. μ = t^T λ/m;
    Calc. μ^aff = (t + α^aff Δt^aff)^T (λ + α^aff Δλ^aff)/m;
    Calc. σ = (μ^aff/μ)^3;
    Calc. residuals with corrector terms: r^corr = (r^x, r^ν, r^λ, r^t + Ω);
    Solve ∇F Δw = −r^corr;
    Calc. α = max{α ∈ (0,1] | (t, λ) + α(Δt, Δλ) ≥ 0} ;
    Update variables: w^+ = w + αβΔw ;
    Calc. residuals used by next predictor step: r;
    if Convergence then
        Terminate algorithm;
```

For the application of model predictive control several methods of time-discretization exits, for nonlinear model predictive control where the model is re-linearized at each temporal point in the prediction horizon, the time-discretization method influences the convergence properties of the NLP solver. Forward Euler time integration is probably the most simple choice. Runge-Kutta time integration schemes are common choices for time discretization of the models. The computation of the sensitivity/Jacobian matrices ($\underline{\mathbf{A}}$,$\underline{\mathbf{B}}$) of the state progress equation can be computationally expensive. Methods such as the one suggested by Kristensen et al. [12], seeks to minimize the computational burden by reusing already calculated information. Yet another time discretization method, which can be used for NMPC is collocation of time-discrete points [13], which shall not be discussed further in this work.

If the models can be assumed to be linear within a time step of the prediction horizon, the computational burden of time-integration might be reduced significantly. The differential state equation and the measured outputs are linearized around $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})$ using first order Taylor series approximation

$$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{A}\boldsymbol{x}_k + \mathbf{B}\boldsymbol{u}_k + \boldsymbol{\delta} \tag{103a}$$

$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \mathbf{C}\boldsymbol{x}_k + \mathbf{D}\boldsymbol{u}_k + \boldsymbol{\gamma} \tag{103b}$$

where the constant contributions $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ are

$$\boldsymbol{\delta} = \boldsymbol{f}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) - \mathbf{A}\bar{\boldsymbol{x}} - \mathbf{B}\bar{\boldsymbol{u}} \tag{104}$$

$$\boldsymbol{\gamma} = \boldsymbol{g}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) - \mathbf{C}\bar{\boldsymbol{x}} - \mathbf{D}\bar{\boldsymbol{u}} \tag{105}$$

The time-discrete state progress equation

$$\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))dt}_{\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k)} \tag{106}$$

can, under the assumption of being linear and having a constant input $\boldsymbol{u}$ within a time step, be time-discretized by the zero-order-hold method [14] and written as

$$\underline{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t)dt + \int_{t_k}^{t_{k+1}} \boldsymbol{\delta}dt = \underline{\mathbf{A}}\boldsymbol{x}_k + \underline{\mathbf{B}}\boldsymbol{u}_k + \underline{\boldsymbol{\delta}} \tag{107}$$

where

$$\underline{\boldsymbol{\delta}} = \Delta t\boldsymbol{\delta} \tag{108}$$

and where

$$\begin{bmatrix} \underline{\mathbf{A}} & \underline{\mathbf{B}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \text{expm}\left(\Delta t \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) \tag{109}$$

where **expm** is the matrix exponential function, usually approximated by a Padé approximation with scaling and squaring [15]. Linear forward Euler time integration can be used instead of the zero-order-hold method. The time step Δt can be divided into n even smaller equidistant time steps and a forward recursion can calculate the state values at the next time step

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \tfrac{\Delta t}{n}\dot{\boldsymbol{x}}_i = \boldsymbol{x}_i + \tfrac{\Delta t}{n}\left(\mathbf{A}\boldsymbol{x}_i + \mathbf{B}\boldsymbol{u}_i + \boldsymbol{\delta}\right) \tag{110}$$

where i is the local time index between time t_k and t_{k+1}. The time discrete state and input matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are initiated as

$$\underline{\mathbf{A}} = \mathbf{I} \text{ and } \underline{\mathbf{B}} = \mathbf{0} \tag{111}$$

and updated as

$$\underline{\mathbf{A}} = (\mathbf{I} + \tfrac{\Delta t}{n}\mathbf{A})\underline{\mathbf{A}} \text{ and } \underline{\mathbf{B}} = (\mathbf{I} + \tfrac{\Delta t}{n}\mathbf{A})\underline{\mathbf{B}} + \tfrac{\Delta t}{n}\mathbf{B} \tag{112}$$

during the n number of time steps, if the time discrete sensitivity matrices are needed.

Notice that the linearized functions (103) are functions of the original variable and not perturbations around the linearization points. If used for NMPC, the iterations within the sequential programming solver uses the perturbations instead of the actual value and a reformulation is required.

6 Control Moves

A typical Model Predictive Control formulation includes penalties on control moves as well as constraints on control moves. A control move $\boldsymbol{u}_k^{\Delta}$ relates to the control signal in the following way

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + \boldsymbol{u}_k^{\Delta} \tag{113}$$

and relates to temporal derivative of the control signal $\boldsymbol{u}_k^{\Delta}/\Delta t = \dot{\boldsymbol{u}}_k$. The system presented in the previous sections can be augmented to handle control moves within the existing framework. Take the extended problem where control moves have been included

$$\min \sum_{k=N}^{\infty} \left(\phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \tfrac{1}{2}\left\|\boldsymbol{u}_k^{\Delta}\right\|^2_{\mathbf{W}_{\boldsymbol{\Delta}}} \right) + \sum_{k=0}^{N-1} \left(\phi_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \tfrac{1}{2}\left\|\boldsymbol{u}_k^{\Delta}\right\|^2_{\mathbf{W}_{\boldsymbol{\Delta}}} \right) + \sum_{k=1}^{N-1} \tfrac{1}{2}\|\boldsymbol{\sigma}_k\|^2_{\mathbf{W}_{\sigma}} \tag{114}$$

subject to (1c) and to the state progress equation (2a) in the interval ($k = 0, \ldots, \infty$) and to the inequality constraints (28b), (28c) and (28d) and to the control move constraint

$$\mathbf{C}_{\boldsymbol{\Delta}}\boldsymbol{u}_k^{\Delta} \leq \boldsymbol{h}_{\boldsymbol{\Delta}}, \qquad k = (0, \ldots, N-1) \tag{115}$$

then by augmenting the state progress equation

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_k \end{bmatrix}}_{\tilde{\boldsymbol{x}}_{k+1}} = \underbrace{\begin{bmatrix} \underline{\mathbf{A}} & \underline{\mathbf{B}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tilde{\underline{\mathbf{A}}}} \underbrace{\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix}}_{\tilde{\boldsymbol{x}}_k} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\tilde{\underline{\mathbf{B}}}} \underbrace{\boldsymbol{u}_k^{\Delta}}_{\tilde{\boldsymbol{u}}_k} + \underbrace{\begin{bmatrix} \underline{\boldsymbol{\delta}} \\ \mathbf{0} \end{bmatrix}}_{\tilde{\underline{\boldsymbol{\delta}}}} \tag{116a}$$

the controlled outputs

$$\tilde{\boldsymbol{g}}_{\boldsymbol{r}}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k) \approx \underbrace{\begin{bmatrix} \mathbf{C}_{\mathbf{r}} & \mathbf{D}_{\mathbf{r}} \end{bmatrix}}_{\tilde{\mathbf{C}}_{\mathbf{r}}} \tilde{\boldsymbol{x}}_k + \underbrace{\begin{bmatrix} \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{D}}_{\mathbf{r}}} \tilde{\boldsymbol{u}}_k + \underbrace{\begin{bmatrix} \boldsymbol{\gamma}_{\boldsymbol{r}} \end{bmatrix}}_{\tilde{\boldsymbol{\gamma}}_{\boldsymbol{r}}} \tag{116b}$$

$$\tilde{\boldsymbol{g}}_{\boldsymbol{z}}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k) \approx \underbrace{\begin{bmatrix} \mathbf{C}_{\mathbf{z}} & \mathbf{D}_{\boldsymbol{z}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{C}}_{\boldsymbol{z}}} \tilde{\boldsymbol{x}}_k + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\tilde{\mathbf{D}}_{\boldsymbol{z}}} \tilde{\boldsymbol{u}}_k + \underbrace{\begin{bmatrix} \boldsymbol{\gamma}_{\boldsymbol{z}} \\ \mathbf{0} \end{bmatrix}}_{\tilde{\boldsymbol{\gamma}}_{\boldsymbol{z}}} \tag{116c}$$

and the inequality constraints

$$\tilde{\boldsymbol{g}}_{\boldsymbol{s}}(\tilde{\boldsymbol{x}}_k) - \boldsymbol{\sigma}_k \approx \underbrace{\begin{bmatrix}\mathbf{C_s} & \mathbf{0}\end{bmatrix}}_{\tilde{\mathbf{C}}_\mathbf{s}} \tilde{\boldsymbol{x}}_k + \underbrace{\begin{bmatrix}\mathbf{0}\end{bmatrix}}_{\tilde{\mathbf{D}}_\mathbf{r}} \tilde{\boldsymbol{u}}_k + \underbrace{\begin{bmatrix}\boldsymbol{\gamma_s}\end{bmatrix}}_{\tilde{\boldsymbol{\gamma}}_s} - \boldsymbol{\sigma}_k \leq \boldsymbol{s} \tag{116d}$$

$$\tilde{\boldsymbol{g}}_{\boldsymbol{h}}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k) \approx \underbrace{\begin{bmatrix}\mathbf{C_h} & \mathbf{D_h} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}}_{\tilde{\mathbf{C}}_\mathbf{h}} \tilde{\boldsymbol{x}}_k + \underbrace{\begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix}}_{\tilde{\mathbf{D}}_\mathbf{h}} \tilde{\boldsymbol{u}}_k + \underbrace{\begin{bmatrix}\boldsymbol{\gamma_h} \\ \mathbf{0}\end{bmatrix}}_{\tilde{\boldsymbol{\gamma}}_h} \leq \underbrace{\begin{bmatrix}\boldsymbol{h} \\ \boldsymbol{h_\Delta}\end{bmatrix}}_{\tilde{\boldsymbol{h}}} \tag{116e}$$

and the dynamic oputput weight matrice, which should also be augmented

$$\tilde{\mathbf{W}}_\mathbf{z} = \begin{bmatrix}\mathbf{W_z} & \mathbf{0} \\ \mathbf{0} & \mathbf{W_\Delta}\end{bmatrix} \tag{117}$$

then the augmented system can be used in the algorithms of the previous sections and the control signal to be send to the plant is

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + \tilde{\boldsymbol{u}}_k \tag{118}$$

If constraints are also present on higher order derivatives of the control signals, then the system should be augmented accordingly.

Bibliography

[1] W.J. Rugh and J.S. Shamma. Research on gain scheduling. *Automatica (UK)*, 36(10):1401–1425, 2000. ISSN 00051098.

[2] Gabriele Pannocchia and James B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2):426–437, 2003. ISSN 00011541.

[3] C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of interior-point methods to model predictive control. *J. Optim. Theory Appl. (USA)*, 99(3):723–757, 1998. ISSN 00223239.

[4] K. Edlund, L.E. Sokoler, and J.B. Jrgensen. A primal-dual interior-point linear programming algorithm for mpc. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 351 –356, dec. 2009. doi: 10.1109/CDC.2009.5400440.

[5] N. Haverbeke, M. Diehl, and B. De Moor. A structure exploiting interior-point method for moving horizon estimation. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 1273 – 1278, Dec. 2009.

[6] Matthew J. Tenny, Stephen J. Wright, and James B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, 2004. ISSN 15732894.

[7] III Arnold, W.F. and A.J. Laub. Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proc. IEEE (USA)*, 72(12):1746–1754, 1984. ISSN 00189219.

[8] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *Automatic Control, IEEE Transactions on*, 43(8):1163 –1169, 1998. ISSN 00189286.

[9] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. Society For Industrial & Applied Mathematics,U.S., 1997.

[10] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[11] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992. doi: 10.1137/0802028. URL `http://link.aip.org/link/?SJE/2/575/1`.

[12] Morten Rode Kristensen, John Bagterp Jørgensen, Per Grove Thomsen, Morten Rode Kristensen, and Sten Bay Jørgensen. An esdirk method with sensitivity analysis capabilities. *Comput. Chem. Eng.*, 28(12):2695–2707, 2004. ISSN 00981354.

[13] Lorenz T. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process.: Process Intensif.*, 46(11):1043–1053, 2007. ISSN 02552701.

[14] Gene F. Franklin, J. David Powell, and Michael Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 2nd edition, 1990.

[15] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM journal on matrix analysis and applications*, 26(4):1179–1193, 2005. ISSN 10957162.

IMM - TECHNICAL REPORT - 2010-14

A Trust-region-based Sequential Quadratic Programming Algorithm

L. C. Henriksen[1] and N. K. Poulsen[2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark, larh@risoe.dtu.dk

[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, nkp@imm.dtu.dk

ABSTRACT

This technical note documents the trust-region-based sequential quadratic programming algorithm used in other works by the authors. The algorithm seeks to minimize a convex nonlinear cost function subject to linear inequality constraints and nonlinear equality constraints.

KEYWORDS

nonlinear programming problem, sequential quadratic programming, trust region

1 Introduction

A nonlinear cost function subject to nonlinear equality and inequality constraints constitutes a nonlinear programming problem (NLP). Multiple methods for solving such problems exists, best known are sequential quadratic programming problem (SQP) [1, 2] algorithms and interior point (IP) algorithms [3, 4] or hybrids of the two [5]. The SQP is typically augmented with a trust region inequality constraint, which can be tightened of loosened depending on the progress made by the current search direction. More advanced versions of these algorithms are able to cope with nonlinear inequality constraints, a feature not yet tested with the algorithm presented in this paper. This paper describes a trust-region-based sequential quadratic programming (TRSQP) algorithm with a general framework which can be used for different problems with adaptations suited for the particular structure of the problem. Such a particular structure is seen in Nonlinear Model Predictive Control [6, 7, 8].

2 Trust-region-based sequential quadratic programming

A NLP of the general form

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}), \text{ s.t. } \boldsymbol{c}(\boldsymbol{x}) = 0 \text{ and } \boldsymbol{d}(\boldsymbol{x}) \leq= 0 \tag{1}$$

has the Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\nu}^T \boldsymbol{c}(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{d}(\boldsymbol{x}) \tag{2}$$

leading to the Karush-Kuhn-Tucker (KKT) conditions for optimality [3, 4]

$$\nabla\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = \nabla \boldsymbol{f}(\boldsymbol{x}) + \nabla \boldsymbol{c}(\boldsymbol{x})\boldsymbol{\nu} + \nabla \boldsymbol{d}(\boldsymbol{x})\boldsymbol{\lambda} = 0 \tag{3}$$

$$\boldsymbol{c}(\boldsymbol{x}) = 0 \tag{4}$$

$$\boldsymbol{d}(\boldsymbol{x}) \leq 0 \tag{5}$$

$$\text{diag}(\boldsymbol{d}(\boldsymbol{x}))\text{diag}(\boldsymbol{\lambda}) = 0 \tag{6}$$

$$\boldsymbol{\lambda} \geq 0 \tag{7}$$

the KKT conditions will not be investigated further here as they more relevant for the IP methods. The NLP can be solved via an iterative procedure known as sequential quadratic programming (SQP) where

in each iteration the local approximated problem is solved as a quadratic programming problem (QP).

$$\min_{\Delta \boldsymbol{x}} \boldsymbol{m}(\Delta \boldsymbol{x}),\ \boldsymbol{m}(\Delta \boldsymbol{x}) = \tfrac{1}{2}\Delta \boldsymbol{x}^T \mathbf{H} \Delta \boldsymbol{x} + \nabla \boldsymbol{f}(\boldsymbol{x})^T \Delta \boldsymbol{x} \tag{8a}$$

subject to

$$\boldsymbol{c}(\boldsymbol{x}) + \nabla \boldsymbol{c}(\boldsymbol{x})^T \Delta \boldsymbol{x} = 0 \tag{8b}$$

$$\boldsymbol{d}(\boldsymbol{x}) + \nabla \boldsymbol{d}(\boldsymbol{x})^T \Delta \boldsymbol{x} \leq 0 \tag{8c}$$

$$\left\| \mathbf{D} \Delta \boldsymbol{x} \right\|_p \leq \delta \tag{8d}$$

The optimization variable is updated by the iterative progress

$$(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})^+ = (\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) + \Delta(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \tag{9}$$

if certain step acceptance criteria are met. The extra inequality constraint (8d), known as a trust-region, where δ is the trust region radius, can be added to aid the convergence. Iterations continue either until the maximum number of allowed iterations are met, if termination criteria are met or until the algorithm fails due e.g. bad numerical handling of the problem solving or because the problem was ill-posed to begin with. The outline of the TRSQP can be seen in Alg. 1.

Algorithm 1: Trust region based sequential quadratic programming solver

```
Set initial values for (x, ν, λ), δ = δ_max and H = ∇²f;
for ITER from 1 to IMAX do
    Calc. trust-region scaling matrix D;
    Solve QP to obtain Δ(x, ν, λ);
    Set trial variables (x, ν, λ)⁺ = (x, ν, λ) + Δ(x, ν, λ);
    Calc. f(x⁺), c(x⁺) and d(x⁺) and their Jacobians;
    Calc. progress measures ρ and γ;
    Update trust region radius δ;
    Update Quasi-Newton approximation of Hessian of Lagrangian H;
    if Step accepted then
        Set (x, ν, λ) = (x, ν, λ)⁺;
    if Convergence then
        Terminate algorithm;
```

The different points of the algorithm are elaborated in the following sections.

2.1 Step acceptance

Step acceptance is depending on two measures of progress as well as inequality feasibility. The actual to predicted cost reduction ratio

$$\rho = \frac{\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^+)}{-\boldsymbol{m}(\Delta \boldsymbol{x})} \tag{10}$$

provides a measure of how well the QP subproblem resembles the properties of the NLP at the current point. For $\rho \approx 1$, the QP and the NLP are in good agreement. For $\rho > 1$, a greater decrease in cost function than predicted by the QP has occurred and for $0 < \rho < 1$ the actual decrease in cost function was not as good as predicted by the QP. For $\rho < 0$, the NLP and the QP are not in agreement of whether the cost function was decreased or increased with the current step. The relative improvement of the cost function

$$\gamma = \frac{\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^+)}{\boldsymbol{f}(\boldsymbol{x})} \tag{11}$$

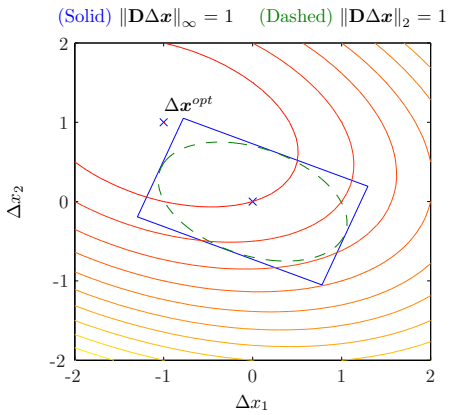


Figure 1: Two trust regions are shown in this figure: The box is given by the ∞-norm and the ellipsoid is given by the 2-norm. Both norms have scaling matrices based on the Hessian.

provides another measure of progress. Different strategies can be taken. In the present work the cost function is allowed to increase as long as ρ or γ are positive and inequalities are feasible. This enables a search where the algorithm is able to move around. The step is accepted if the QP was able to find a feasible solution and if

$$\max(\rho, \gamma) > 0 \tag{12}$$

$$\max(\boldsymbol{d}(\boldsymbol{x}^+)) \leq \tau_{d(x)} \tag{13}$$

furthermore steps are not accepted in the first iteration as experience has shown better performance by letting the algorithm start up and generate its first Quasi-Newton Hessian approximation etc.

2.2 Trust region

A trust region with the general form

$$\|\mathbf{D}\Delta\boldsymbol{x}\|_p \leq \delta \tag{14}$$

where $\mathbf{D}$ is the scaling matrix, p is the number of the norm, e.g. 1,2 or ∞, and δ is the trust region radius, can be imposed on the QP. The first choice of a suitable scaling matrix $\mathbf{D}$ might be the identity matrix. An even better choice takes the Hessian $\mathbf{H}$ into account as the problem might not be equally sensitive to changes of $\boldsymbol{x}$ in all directions. The 2-norm yields a quadratic constraint $\Delta\boldsymbol{x}^T\mathbf{D}^T\mathbf{D}\Delta\boldsymbol{x} \leq \delta^2$ which can be chosen to be identical to the Hessian $\Delta\boldsymbol{x}^T\mathbf{H}\Delta\boldsymbol{x} \leq \delta^2$. The quadratic constraints would make the approximated problem a quadratic constrained quadratic cost (QCQC) problem which does not fit into the normal QP framework used by the SQP, where only linear constraints occur. The ∞-norm results in linear inequality constraints

$$-\delta \leq \mathbf{D}\Delta\boldsymbol{x} \leq \delta \tag{15}$$

and is thus suitable for the description of a trust region which can be included in standard QP. Inspired by the quadratic constraints the Hessian can be decomposed by e.g. singular value decomposition

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}\mathbf{V}^T = \mathbf{D}^T\mathbf{D} \tag{16}$$

giving a multidimensional box circumscribing the ellipsoid of the quadratic constraint as tight as possible. Fig. 1 shows how the ∞-norm and the 2-norm trust regions resemble the Hessian and ensure that steps are constrained along the dimensions in a fashion scaled by the Hessian.

2.3 Trust region radius

The trust region radius should be increased or decreased according to a set of rules: If ρ is small indicating poor agreement between the NLP and QP or if γ is negative indicating an increase in cost function or if the previous step has been rejected for some reason, then the trust region radius should be decreased. If the QP failed, it is most likely due to a too restrictive trust region radius and the radius should be increased. If ρ was high, indicating good agreement between the NLP and the QP, then the trust region radius should be increased.

$$\delta = \begin{cases} \delta/2 & \text{for} \quad \rho < 1/4 \text{ or } \gamma < 0 \text{ or Step Rejected} \\ \min(3\delta, \delta_{max}) & \text{for} \quad \text{QP failed} \\ \min(3\min(\delta, \|\mathbf{D}\Delta\boldsymbol{x}\|_p), \delta_{max}) & \text{for} \quad \rho > 3/4 \end{cases} \tag{17}$$

2.4 Quasi-Newton approximations of Hessian of Lagrangian

The quadratic cost in the QP should ideally by the Hessian of the Lagrangian of the NLP

$$\nabla^2\mathcal{L} = \nabla^2\boldsymbol{f}(\boldsymbol{x}) + \sum_i^{n_c} \nabla^2\boldsymbol{c}_i(\boldsymbol{x})\nu_i + \sum_i^{n_d} \nabla^2\boldsymbol{d}_i(\boldsymbol{x})\lambda_i \tag{18}$$

this would computational expensive and if no analytic second derivatives of the cost function and constraints are available, those would have to be determined via finite differences. Commonly used alternatives are different Quasi-Newton approximations such as dBFGS (Alg. 2), SR1 (Alg. 3) and SR1pos (Alg. 4). They are all calculations based on the variable step $\boldsymbol{s}$ and and with the difference in the gradients of the Lagrangian $\boldsymbol{y}$

$$\boldsymbol{s} = \boldsymbol{x}^+ - \boldsymbol{x} \tag{19}$$

$$\boldsymbol{y} = \nabla\mathcal{L}(\boldsymbol{x}^+, \boldsymbol{\nu}^+, \boldsymbol{\lambda}^+) - \nabla\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}^+, \boldsymbol{\lambda}^+) \tag{20}$$

Algorithm 2: damped Broyden-Fletcher-Goldfarb-Shanno (dBFGS) update

if $\boldsymbol{s}^T\boldsymbol{y} < 0.2\boldsymbol{s}^T\mathbf{H}\boldsymbol{s}$ **then**
  $\theta = 0.8\boldsymbol{s}^T\mathbf{H}\boldsymbol{s}(\boldsymbol{s}^T\mathbf{H}\boldsymbol{s} - \boldsymbol{s}^T\boldsymbol{y})^{-1}$;
  $\boldsymbol{y} = \theta\boldsymbol{y} + (1-\theta)\mathbf{H}\boldsymbol{s}$;
$\mathbf{H} = \mathbf{H} - \mathbf{H}\boldsymbol{s}\boldsymbol{s}^T\mathbf{H}(\boldsymbol{s}^T\mathbf{H}\boldsymbol{s})^{-1} + \boldsymbol{y}\boldsymbol{y}^T(\boldsymbol{y}^T\boldsymbol{s})^{-1}$;

Algorithm 3: Symmetric rank-1 (SR1) update

if $(\boldsymbol{y} - \mathbf{H}\boldsymbol{s}) < 10^{-6}(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})\boldsymbol{s}$ **then**
  $\mathbf{H} = \mathbf{H} + (\boldsymbol{y} - \mathbf{H}\boldsymbol{s})(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T((\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T\boldsymbol{s})^{-1}$;

Algorithm 4: Positive definite symmetric rank-1 (SR1pos) update

if $(\boldsymbol{y} - \mathbf{H}\boldsymbol{s}) < 10^{-6}(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})\boldsymbol{s}$ **then**
  $\mathbf{H} = \mathbf{H} + (\boldsymbol{y} - \mathbf{H}\boldsymbol{s})(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T((\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T\boldsymbol{s})^{-1}$;
  $\boldsymbol{\Lambda}\mathbf{V} = \mathbf{H}\mathbf{V}$;
  $\mathbf{H} = \mathbf{V}\boldsymbol{\Lambda}^+\mathbf{V}^{-1}$;

The dBFGS and SR1pos both maintain a positive definite Hessian, making the QP easier to solve. The SR1 might be a better fit if the NLP is not positive definite.

2.5 Termination

The algorithm terminates if steps toward the optimum are becoming to small and if the last accepted step was feasible

$$\|\Delta \boldsymbol{x}\|_2 \leq \tau_{\Delta x} \tag{21}$$

$$\max(\boldsymbol{d}(\boldsymbol{x}^+)) \leq \tau_{d(x)} \tag{22}$$

$$\left\|\boldsymbol{c}(\boldsymbol{x}^+)\right\|_\infty \leq \tau_{c(x)} \tag{23}$$

3 Discussion and future work

The presented algorithm has been tested with Nonlinear Model Predictive Control, with nonlinear equality constraints [6]. It has not been tested with nonlinear inequality constraints and performance with nonlinear inequality constraints remains to be investigated.

References

[1] Richard H. Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):–, 2000.

[2] Stephen J. Wright and Matthew J. Tenny. A feasible trust-region sequential quadratic programming algorithm. *SIAM journal on optimization*, 14(4):1074–1105, 2004.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] R.A. Waltz, J. Nocedal, D. Orban, and J.L. Morales. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.*, 107(3):391–408, 2006.

[6] L. C. Henriksen, N. K. Poulsen, and M. H. Hansen. Nonlinear model predictive control of a simplified wind turbine. *Submitted for Proc. World Congr. Int. Fed. Autom. Control., IFAC*, pages –, 2011.

[7] L. C. Henriksen and N. K. Poulsen. An online re-linearization scheme suited for model predictive or linear quadratic control. IMM-Technical Report 2010-13, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, 2010.

[8] Matthew J. Tenny, Stephen J. Wright, and James B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, 2004.